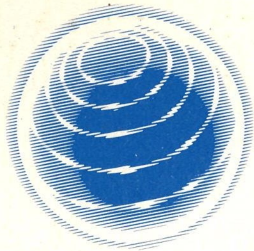


PHILIPS



PHILIPS MSX THUISCOMPUTER VG8020



GEBRUIKSAANWIJZING

MSX

GEBRUIKS-
AANWIJZING

voor de

PHILIPS

MSX-thuiscomputer

VG 8020

INHOUD

VOORWOORD

De Philips MSX-computer is een moderne en veelzijdige thuiscomputer, die u onder meer kunt gebruiken om te leren programmeren en om uw eigen programma's te schrijven. Dat kunnen heel eenvoudige, maar ook erg ingewikkelde programma's zijn. U kunt uw MSX-computer ook gebruiken voor het spelen van een groot aantal verschillende spellen, die u kant-en-klaar kunt kopen.

De MSX-computer heet zo omdat hij voldoet aan de nieuwste norm voor thuiscomputers: de MSX-norm. Het grote voordeel daarvan is dat u gebruik kunt maken van alle programma's uit de omvangrijke MSX- bibliotheek. Een ander voordeel is dat u een ruime keus hebt uit randapparaten, zoals printers, cassetterecorders en diskettestations.

Uw MSX-computer heeft een ingebouwde MSX-BASIC-interpreter (vertaler), die identiek is aan de BASIC-interpreters van andere MSX-computers.
MSX-BASIC herkent de meeste instructies van Microsoft's MBASIC-80, maar kent ook instructies voor muziek, kleur en bewegende beelden ("sprites") en het gebruik van spelregelaars ("joy sticks").

In deze gebruiksaanwijzing beginnen we met de instructies voor het in gebruik nemen van uw MSX-computer.

Hoofdstuk 2 geeft een beschrijving van een aantal MSX-BASIC-instructies die u kunt gebruiken. Niet alle instructies konden in deze gebruiksaanwijzing worden behandeld. De gebruiksaanwijzing is dan ook niet bedoeld als een cursus programmeren in MSX-BASIC. Daarvoor verwijzen wij u naar de erkende instellingen voor schriftelijk onderwijs, die cursussen "Programmeren in MSX-BASIC" in hun pakket hebben, en naar de literatuur over MSX-BASIC in de boekhandel.

Een alfabetische opsomming van alle instructies kunt u vinden in het "Naslagwerk voor programmeurs" ("Reference Manual").

Hoofdstuk 3 beschrijft hoe u de mogelijkheden van uw computer kunt uitbreiden met behulp van extra apparatuur.

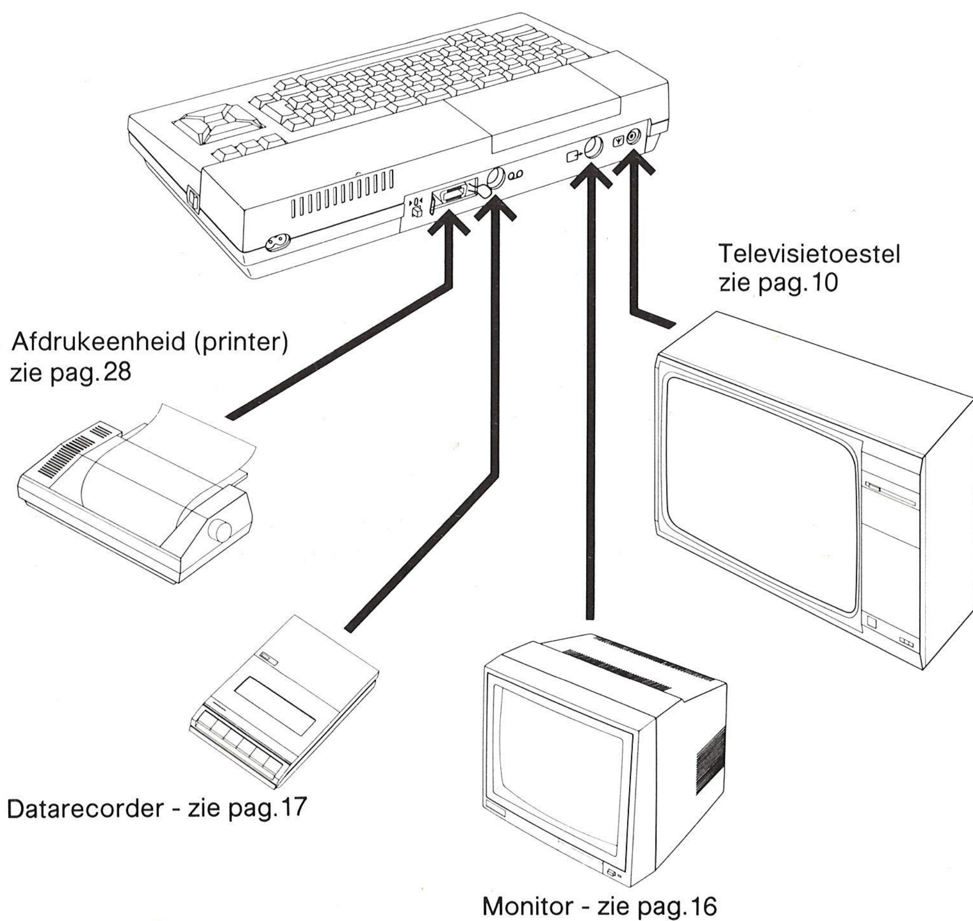
Hoofdstuk 4 tenslotte bestaat uit een aantal appendices (aanhangsels) met belangrijke aanvullende informatie.

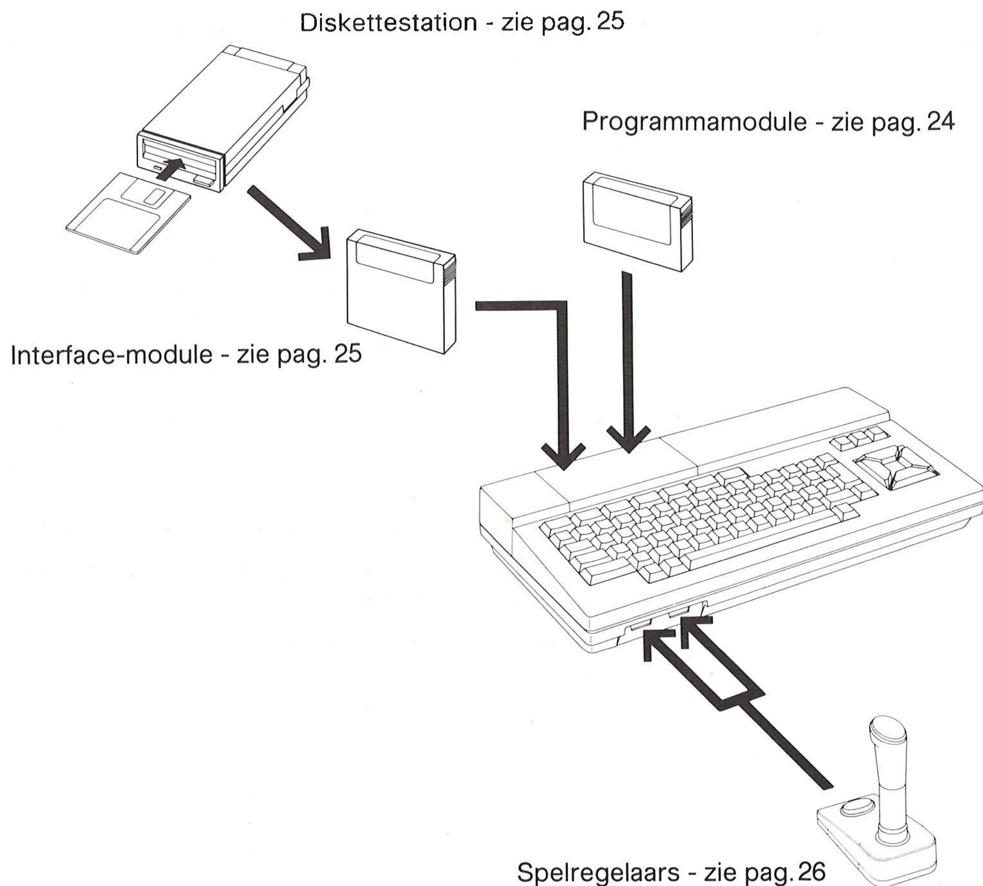
Uw MSX-computer heeft een vrij programmeerbaar geheugen (RAM) van 64 Kbyte. Dit geheugen is alleen beschikbaar als u geen gebruik maakt van de ingebouwde MSX-BASIC-interpreter, die 32 Kbyte geheugenruimte in beslag neemt. Bij gebruik van de BASIC-interpreter wordt deze in zijn geheel gekopieerd naar het RAM-geheugen, waardoor u circa 32 Kbyte vrije geheugenruimte ter beschikking houdt voor programma's, gegevens en dergelijke. Er zijn programma's op cassette of diskette die bij de uitvoering geen gebruik maken van de BASIC-interpreter. Dergelijke programma's kunnen beschikken over ongeveer 60 Kbyte RAM-geheugen.

Een advies tot besluit: lees deze handleiding zorgvuldig en probeer alle gegeven voorbeelden. U zult dan al gauw uw eigen programma's kunnen maken.

1 KENNISMAKING MET UW MSX-COMPUTER

1.1 De MSX-computer en de randapparaten





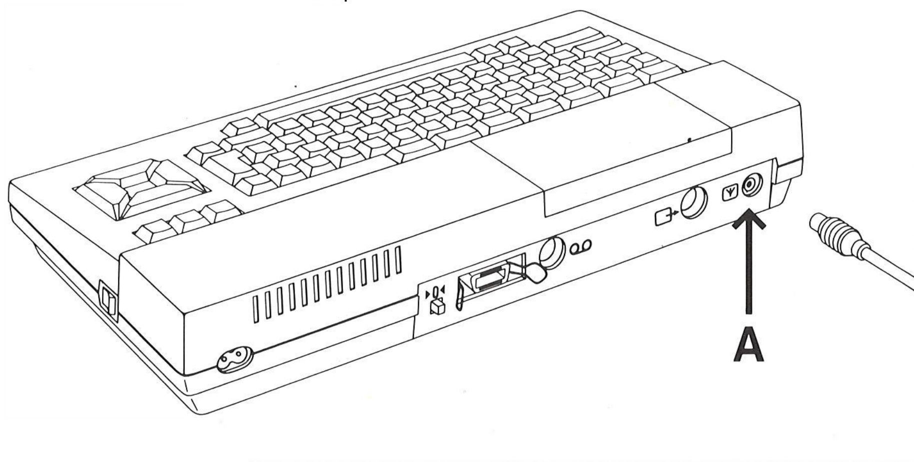
1.2 Het maken van de aansluitingen

In de doos vindt u alles dat u nodig hebt om in combinatie met een televisietoestel met uw nieuwe MSX-computer te kunnen werken, inclusief een kabel om uw MSX-computer te verbinden met de antenne-aansluiting van uw TV-toestel.

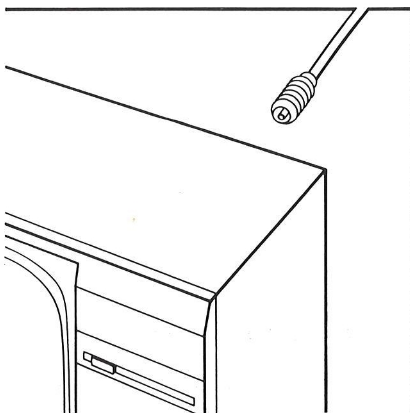
1.2.1 Standaard-opstelling

Het installeren van de computer is erg eenvoudig, als u de volgende stappen neemt:

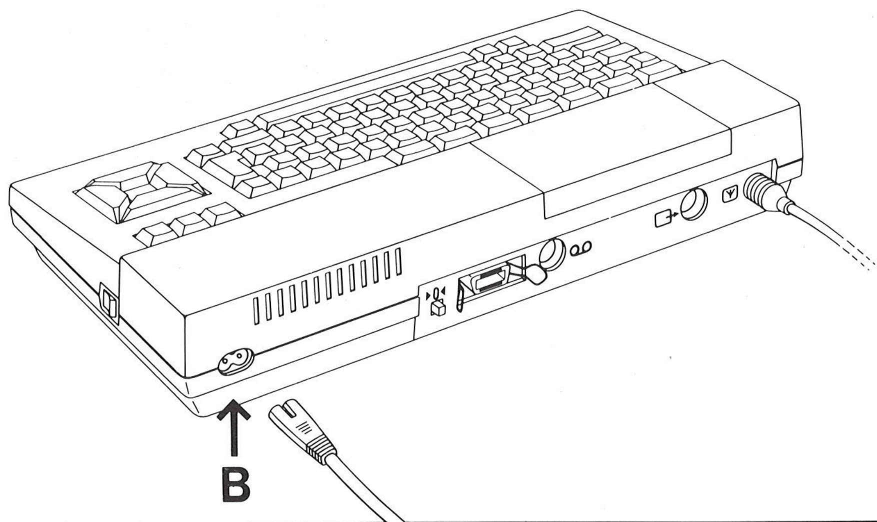
Verbind de antennekabel met de TV-aansluiting (A) van uw computer.



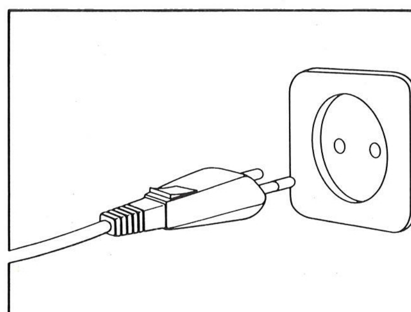
Verbind het andere einde van de kabel met de antenne-ingang van uw TV-toestel.



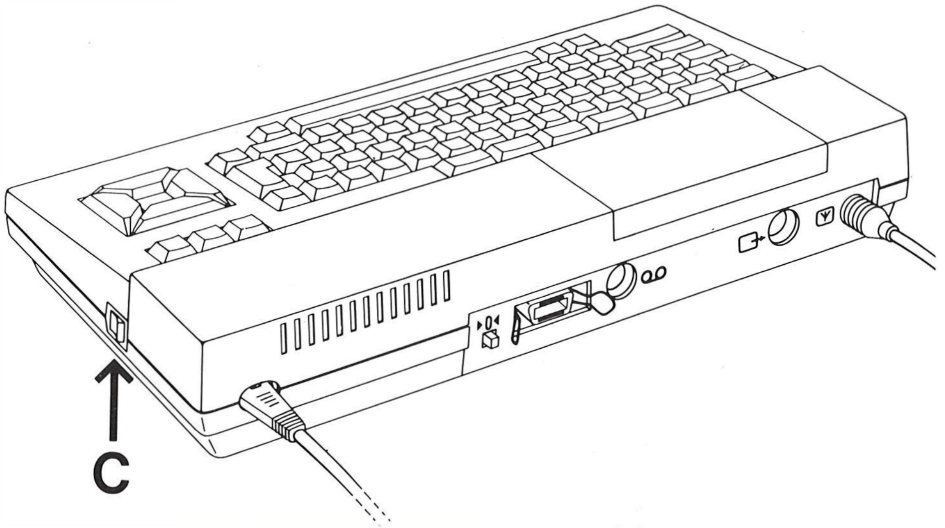
Sluit het netsnoer aan op de netaansluiting (B) van uw computer.



Steek de steker van het netsnoer in het stopcontact.



Schakel de computer in door de aan-uitschakelaar (C) in te drukken.



Schakel het TV-toestel in.

Stem het TV-toestel af op het juiste UHF-kanaal. Het kanaal staat aangegeven op de achterkant van de computer, naast de TV-aansluiting (A). Gebruikt u het TV-toestel ook om TV-programma's te ontvangen, dan verdient het aanbeveling één van de voorkeuzetoetsen te reserveren voor uw computer.

Als het TV-toestel op het juiste kanaal is afgestemd, moet u de volgende tekst op het scherm zien:

```
MSX BASIC
enz.
```

In plaats van "enz." staat er tekst op het scherm die aangeeft wie het auteursrecht van MSX-BASIC bezit en hoeveel vrije geheugenruimte er beschikbaar is. Deze teksten kunnen verschillen, daarom hebben we ze hier niet afgedrukt.

1.3 De eerste oefening

Als u serieus uw eigen computerprogramma's wilt gaan schrijven, zult u wat tijd moeten besteden aan het bestuderen van deze gebruiksaanwijzing. Maar u hoeft natuurlijk niet alles van programmeren af te weten om alvast een aardig klein programma in te toetsen.
We nemen aan dat u in elk geval enigszins overweg kunt met een schrijfmachinetoetsenbord. Dan zal het u opvallen dat het toetsenbord van de computer veel overeenkomst vertoont met dat van een schrijfmachine en dat de lettertoetsen op dezelfde plaatsen zitten. De computer heeft een aantal toetsen extra; die zullen later in deze gebruiksaanwijzing ter sprake komen.
Voordat u het voorbeeldprogramma gaat intikken moeten we even wat meer over enkele toetsen vertellen. Er zijn twee hoofdlettertoetsen op de één na onderste rij, gemerkt met ↑. Deze toetsen zullen we in deze gebruiksaanwijzing aanduiden met SHIFT. Drukt u een SHIFT-toets in, dan krijgt u hoofdletters op het scherm. Ook om aanhalingstekens, het dollarteken en dergelijke op het scherm te krijgen moet u een SHIFT-toets indrukken.
De grote toets rechts, gemerkt met een gebroken pijl, is de regelterugtoets. Deze zullen we verder aanduiden met RETURN-toets.

Als u de geluidssterkteregelaar van uw TV-toestel wat hoger zet, zult u horen dat bij het indrukken van de meeste toetsen een klik uit de luidspreker klinkt.

Eerst nog een belangrijk advies: maak er een goede gewoonte van zeer secuur te werken. Elke afwijking bij het intikken van de instructies kan tot gevolg hebben dat het programma niet goed werkt of dat een foutmelding op het scherm verschijnt.

Tik nu de volgende regels in. Gebruik de spatiebalk om ruimte tussen de woorden en symbolen te krijgen, precies zoals in het voorbeeld is gedaan. Sluit elke regel af met het indrukken van de RETURN-toets en ga daarna verder met de volgende regel.

```
10 A$="MSX thuiscomputer    "
20 PRINT A$;
30 GOTO 20
```

Zie nu wat er gebeurt als u functietoets F5 (bovenste rij toetsen) indrukt. De computer blijft eindeloos dezelfde woorden op het scherm schrijven, totdat u op de STOP-toets drukt. Drukt u nogmaals op STOP, dan gaat de computer verder waar hij gebleven was.

Om de uitvoering van een programma definitief af te breken, moet eerst de CTRL-toets ingedrukt worden gehouden en daarna de STOP-toets worden ingedrukt. U ziet dan het volgende bericht op het scherm:

```
Break in 20
```

of

```
Break in 30
```

De getallen 20 en 30 geven aan in welke programmaregel de uitvoering van het programma werd afgebroken.

Wilt u het programma nog eens bekijken, druk dan functietoets F4 in, gevolgd door RETURN. Dan ziet u het volgende op het scherm:

```
LIST
10 A$="MSX thuiscomputer    "
20 PRINT A$;
30 GOTO 20
Ok
```

Als wat u op het scherm ziet, niet overeenstemt met ons voorbeeld, hebt u een fout gemaakt. Bedenk dat een kleine vergissing al tot een heel ander resultaat kan leiden. U kunt zelfs een "Syntax error" op het scherm krijgen. Dit betekent dat u hebt gezondigd tegen de syntaxis, dat wil zeggen de taalregels, van de taal MSX-BASIC en dat de computer niet begrijpt wat u bedoelt. Is dit het geval, controleer het programma dan en tik zo

nodig een of meer regels opnieuw in. Vooral de leestekens zijn erg belangrijk. Een dubbele punt heeft bij voorbeeld voor MSX-BASIC een heel andere betekenis dan een puntkomma.

1.4 Het gebruik van een monitor

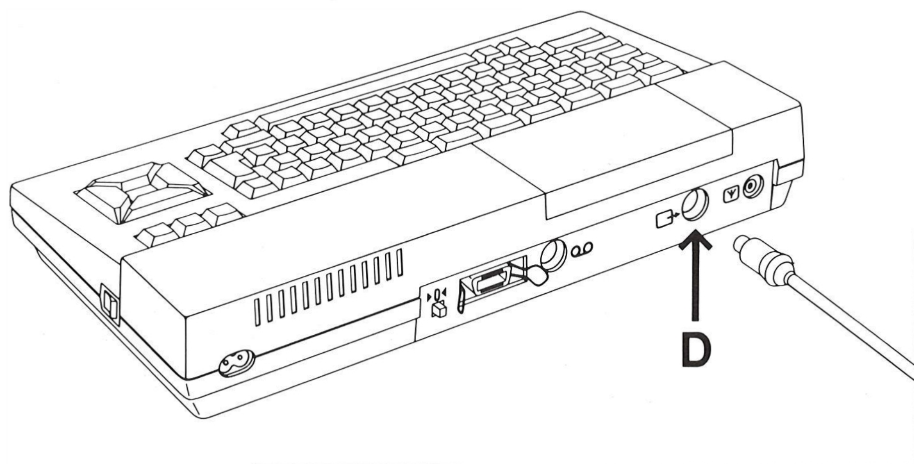
Een monitor is een toestel met een beeldscherm, dat speciaal is ontworpen voor gebruik met computers.

Een monochrome monitor geeft beelden in één kleur, meestal in tinten van groen omdat dit een rustgevende kleur is. Er zijn ook monochrome monitors met een amberkleurig beeld, dat een beter contrast heeft. Er wordt gebruik gemaakt van fosfors die een langere nalichttijd hebben dan de fosfors die voor TV-beeldschermen worden gebruikt. Daardoor wordt een rustig beeld verkregen, zonder flikkeringen, dan minder vermoeiend is voor de ogen. Vooral voor het weergeven van tekst heeft een monochrome monitor grote voordelen.

Er zijn ook kleurenmonitors in de handel die speciaal zijn ontwikkeld voor gebruik met computers. Ook deze hebben fosfors met een langere nalichttijd dan TV-beeldschermen en geven dus een rustig, stabiel en flikkervrij beeld. Kleur is uiteraard belangrijk als u de grafische mogelijkheden van uw MSX-computer volledig wilt uitbuiten.

Veel moderne TV-toestellen zijn, met het oog op de snelle opkomst van thuiscomputers, uitgerust met een zogenaamde Euro- of SCART-aansluiting. Deze maakt het mogelijk het toestel te gebruiken alsof het een monitor is. Het voordeel daarvan is dat het beeldsignaal niet eerst in de computer wordt gemoduleerd en in het TV-toestel weer wordt gedemoduleerd, waardoor kwaliteitsverlies optreedt. Als uw computer op deze video-aansluiting wordt aangesloten is de beeldkwaliteit beter dan wanneer hij op de antenne-ingang wordt aangesloten.

Uw MSX-computer heeft een speciale uitgang voor monitors (D). Er is een speciale kabel voor nodig. Welke dat is, hangt af van het type monitor dat u gebruikt. Vraag uw handelaar.



1.5 Het gebruik van een datarecorder

Als u een computerprogramma schrijft, wordt dit tijdelijk opgeslagen in het werkgeheugen van de computer. Op het moment dat u de computer uitschakelt, gaat de inhoud van het geheugen en daarmee uw programma verloren. Een datarecorder maakt het u mogelijk uw programma's vast te leggen op de cassetteband, zelfs als ze nog niet klaar zijn. U kunt ze op elk gewenst moment opnieuw inlezen in het geheugen van de computer, bij voorbeeld om er verder aan te werken.
Een datarecorder is ook onmisbaar als u uw computer wilt gebruiken voor het bewaren van gegevens, zoals boekhoudgegevens, een adressenbestand of een ledenadministratie.
Ook als u van tijd tot tijd een videospel wilt spelen hebt u een datarecorder nodig omdat veel van deze spellen alleen beschikbaar zijn op cassette.

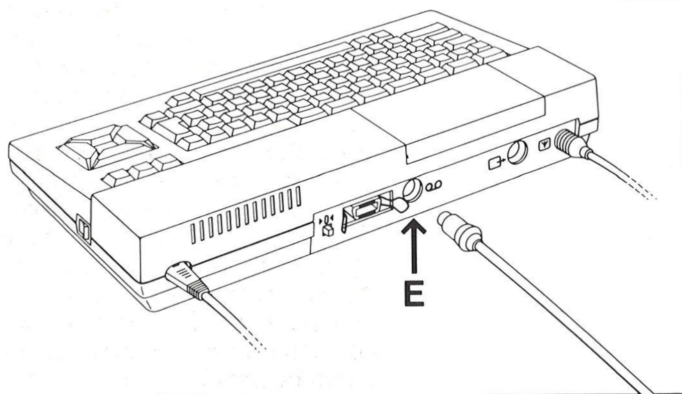
Hoewel u voor dit doel elke goede cassetterecorder kunt gebruiken (en dat hoeft echt geen dure HiFi-recorder te zijn), verdient een datarecorder met afstandbediening ("Remote control") de voorkeur omdat deze tot op zekere hoogte door de computer wordt bestuurd.

1.5.1 Cassettes

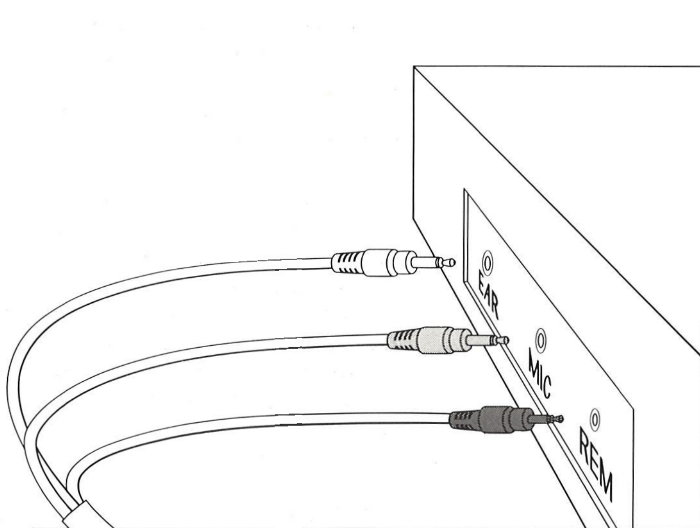
Voor het vastleggen van thuiscomputerprogramma's kunnen cassettes met een "ferro"-band van goede kwaliteit worden gebruikt. Wij bevelen echter het gebruik van speciale computercassettes aan. Het verschil tussen deze en andere cassettes is dat de band minder lang is en dat ze geen aanloopstrook hebben (waarop niet kan worden opgenomen). U hoeft dan niet telkens de band een eindje door te spoelen totdat de aanloopstrook de kop is gepasseerd. De bandlengte van computercassettes is ruim voldoende om een indrukwekkend computerprogramma vast te leggen en deze cassettes zijn doelmatiger in het gebruik als u beide kanten van de cassette wilt gebruiken.
Sommige computercassettes hebben een beveiliging tegen het per ongeluk overschrijven van een programma.

1.5.2 Het aansluiten van een datarecorder

Verbind de DIN-steker met de recorderaansluiting aan de achterkant van de computer (E).



Aan de andere kant van de kabel zijn drie stekers gemonteerd. Steek de zwarte in de aansluiting voor de afstandbediening (meestal aangeduid met REM van REMote control), de rode in de microfoonaansluiting (MIC) en de witte in de oortelefoonaansluiting (aangeduid met EAR of Line Out).



Opmerking:
Voor het aansluiten van een datarecorder hebt u een speciale kabel nodig. Meestal ziet deze eruit zoals we hierboven hebben beschreven. Het verdient echter aanbeveling eerst de gebruiksaanwijzing van de recorder te raadplegen.

1.5.3 Het opslaan van een programma op cassette

Als de datarecorder op de juiste manier op de computer is aangesloten, kunt u hem gebruiken om informatie op te slaan en deze later weer teruglezen in het computergeheugen.
Wij raden u aan hiermee een beetje te oefenen om uzelf vertrouwd te maken met het gebruik van de datarecorder. U kunt oefenen met het programma dat u eerder hebt gemaakt:

```
10 A$="MSX thuiscomputer  "
20 PRINT A$;
30 GOTO 20
```

Druk tegelijk de opneem- en de afspeeltoets (REC en PLAY) van de recorder is, alsof u een gewone geluidsopname wilt maken. U zult zien dat de recorder nog niet gaat lopen. Hij wacht op instructies van de computer. Tik nu in:

```
CSAVE "MSX"
```

"MSX" is een willekeurige naam voor dit miniprogramma. Elk programma dat u op cassette vastlegt moet een naam hebben, zodat de computer het kan terugvinden als u het later weer wilt inlezen. Dit is vooral van belang als u twee of meer programma's op dezelfde cassette wilt opslaan.

Druk nu op RETURN. U zult zien dat de recorder begint op te nemen. Hij zal automatisch stoppen als het programma helemaal is opgenomen. Dat wordt bevestigd met "Ok" op het scherm.

Opmerking 1:
Als u een gewone cassetterecorder zonder afstandbediening gebruikt, dient u als volgt te handelen. Type eerst het commando CSAVE "MSX" in, maar druk nog niet op RETURN. Druk daarna de toetsen REC en PLAY van de recorder in en druk dan pas de RETURN-toets in. Hetzelfde geldt als u wel een datarecorder bezit, maar niet de juiste aansluitkabel met steker voor de afstandbediening hebt.

Opmerking 2:
Als u een gewone cassette met aanloopstrook gebruikt, moet u eerst met PLAY de band een eindje doorspoelen, totdat de aanloopstrook de kop van de recorder is gepasseerd.

Opmerking 3:
Volume- en toonregelaar van de recorder moeten juist zijn ingesteld. Raadpleeg de gebruiksaanwijzing van de

recorder en maak eerst enkele proefopnamen. De volumeregelaar moet meestal ergens tussen de middenstand en de hoogste stand staan.

1.5.4 Controle van het opgenomen programma

Door het opnemen van een programma op cassette (of diskette) wordt het computergeheugen niet gewist. Daadoor is het mogelijk het opgenomen programma te controleren voordat u het geheugen wist, en dat verdient dan ook aanbeveling. Om het programma te controleren spoelt u de cassette terug naar het begin. Druk dan de toets PLAY van de recorder in en tik:

```
CLOAD? "MSX"
```

Druk daarna op RETURN. De computer zal nu het programma dat is vastgelegd op cassette vergelijken met het oorspronkelijke programma in het geheugen van de computer. Als beide programma's identiek zijn zal de computer dit bevestigen met "Ok".

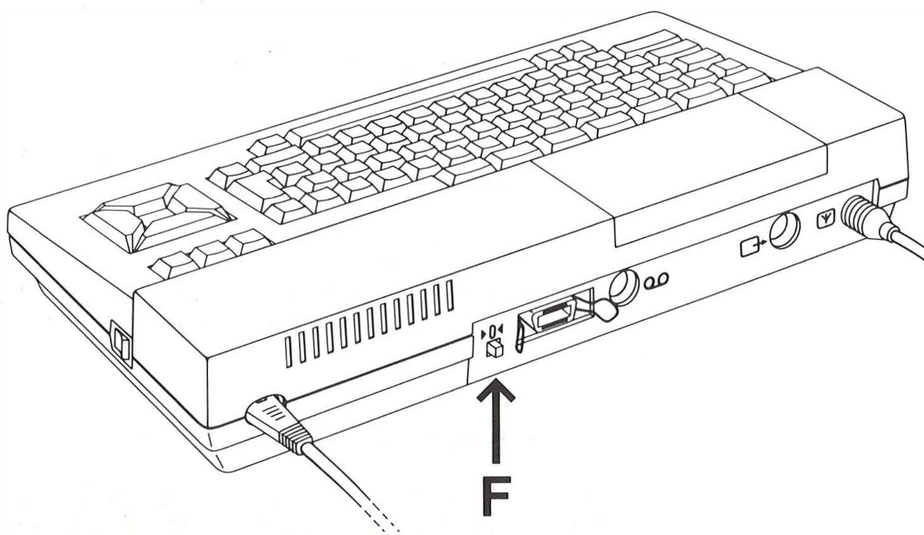
Zijn de programma's bij uitzondering niet identiek, dan verschijnt op het scherm de mededeling:

```
Verify error
```

Zo'n verificatiefout kan het gevolg zijn van een onjuiste instelling van volume- of toonregelaar. Stel deze opnieuw in, spoel de cassette terug en neemt het programma nogmaals op met behulp van het commando CSAVE, gevolgd door de programmanaam tussen aanhalingstekens.

1.5.5 Inlezen van een programma van cassette

Om te kijken of het opnemen goed werkt, raden wij u aan de "reset"-toets (F) aan de achterkant van de computer in te drukken of de computer uit en weer in te schakelen. In beide gevallen wordt het programma uit het computergeheugen gewist en verschijnt het welkomstbeeld weer op het scherm.



Stop de cassette met het programma dat u wilt inlezen in de datarecorder, spoel de cassette helemaal terug en druk de PLAY-toets van de recorder in. Tik dan in:

```
CLOAD "MSX"
```

Druk daarna op RETURN. Het programma zal nu in het computergeheugen worden gelezen.

Opmerking 1:
Als u een gewone cassetterecorder zonder afstandbediening gebruikt, dient u eerst het commando CLOAD "MSX" in te toetsen, daarna op RETURN te drukken en pas dan de recorder te laten lopen door de PLAY-toets in te drukken.

Opmerking 2:
Als het programma dat u wilt inlezen aan het begin van de cassetteband staat, kunt u de naam van het programma weglaten. U toetst dan alleen CLOAD in. De computer zal nu het eerste programma inlezen dat hij tegenkomt. Na korte tijd zal hij dit melden met:

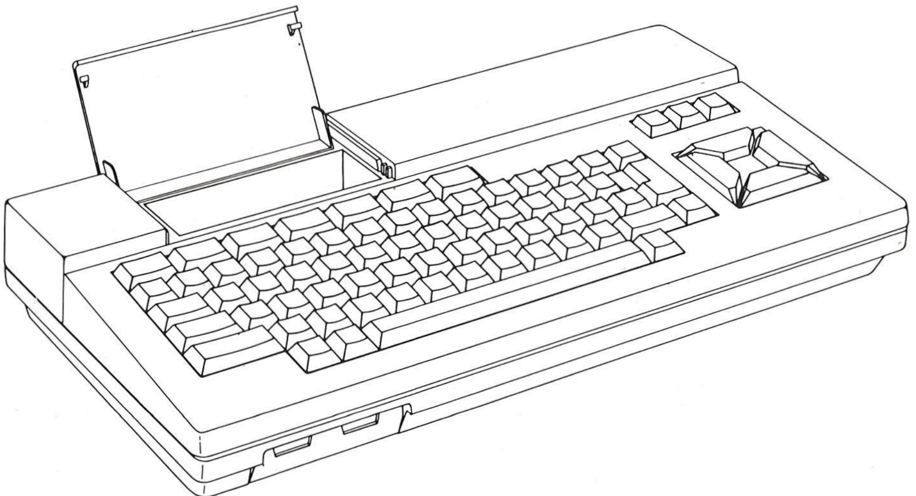
```
FOUND: MSX
```

Als het programma is ingelezen, stopt de datarecorder en verschijnt "Ok" op het scherm.
U hebt nu drie mogelijkheden:

1 Druk functietoets F4 in, gevolg door RETURN. De lijst (Engels "LIST") van programmaregels verschijnt nu op het scherm.
2 Druk functietoets F5 in ("RUN"). Het programma zal dan worden uitgevoerd.
3 U kunt het programma veranderen of aanvullen door nieuwe programmaregels in te tikken. Daarna kunt u het gewijzigde programma op cassette bewaren met het commando CSAVE.

1.6 Het gebruik van insteekmodules

Uw MSX-computer is uitgerust met twee "sleuven" voor insteekmodules, afgedekt door een doorzichtige klep. De aansluitblokken (connectors) worden zichtbaar als u de klep opent. Voor sommige toepassingen is het niet voldoende de klep te openen, maar moet deze helemaal worden verwijderd. Dat kan gemakkelijk gebeuren door hem helemaal achterover te drukken.

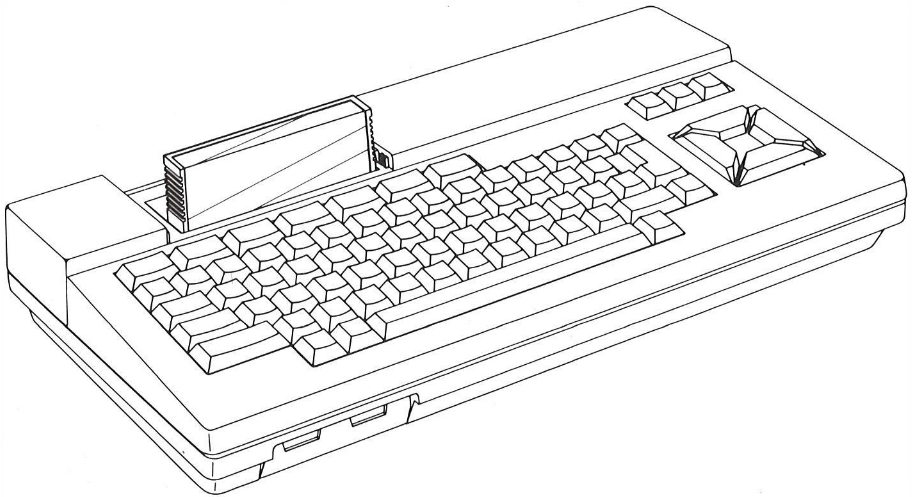


WAARSCHUWING:
SCHAKEL ALTIJD DE COMPUTER UIT VOORDAT U EEN RANDAPPARAAT AANSLUIT OF EEN MODULE IN EEN VAN DE TWEE SLEUVEN STEEKT.
Bedenk hierbij dat alle informatie in het computergeheugen verloren gaat op het moment dat u het apparaat uitschakelt. Sluit daarom alles aan dat u wilt gebruiken voordat u de computer inschakelt.

U kunt de "sleuven" voor een aantal doeleinden gebruiken. Het maakt geen verschil welke van de twee u voor een bepaalde insteekmodule gebruikt, want ze zijn volkomen identiek. Soms zult u beide sleuven tegelijk moeten gebruiken.

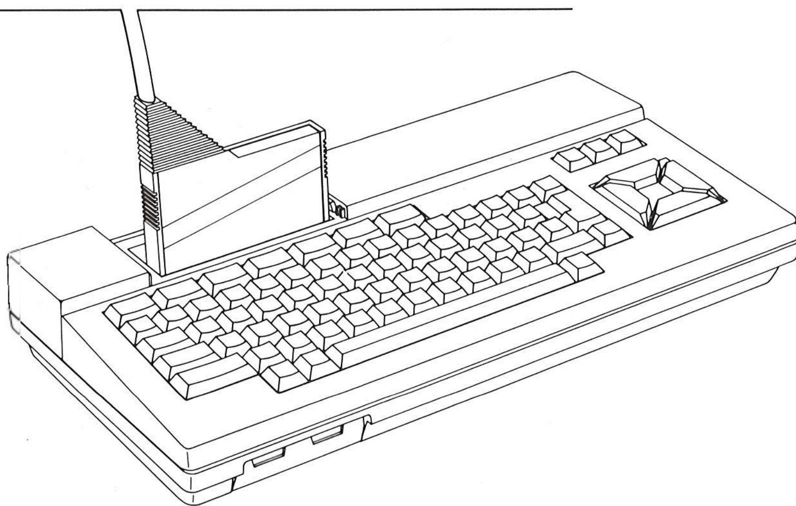
1.6.1 Programmamodules

Een aantal MSX-spelprogramma's, maar ook programma's voor tekst- of bestandsverwerking, is beschikbaar in de vorm van insteekmodules. U kunt deze programma's gebruiken door de modules in één van de twee sleuven te steken, met het etiket naar u toe. Voor informatie over de mogelijkheden en toepassingen van deze programma's verwijzen wij naar de meegeleverde gebruiksaanwijzingen daarvan.

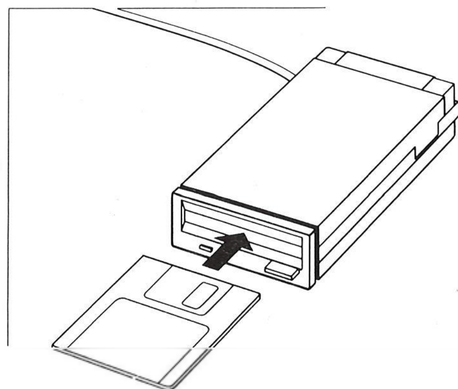


1.6.2 "Interface"-modules

Als u op uw computer een bijzonder randapparaat wilt aansluiten, bij voorbeeld een diskettestation of een modem (modulator-demodulator, nodig om via de telefoon verbinding te maken met andere computers), dan hebt u een speciale "interface"-module nodig (een interface is in het algemeen een schakeling die twee apparaten aan elkaar koppelt). Deze interface-module moet in een van de twee sleuven worden gestoken. Voor nadere bijzonderheden verwijzen wij naar de gebruiksaanwijzingen die bij de interface-modules worden geleverd.

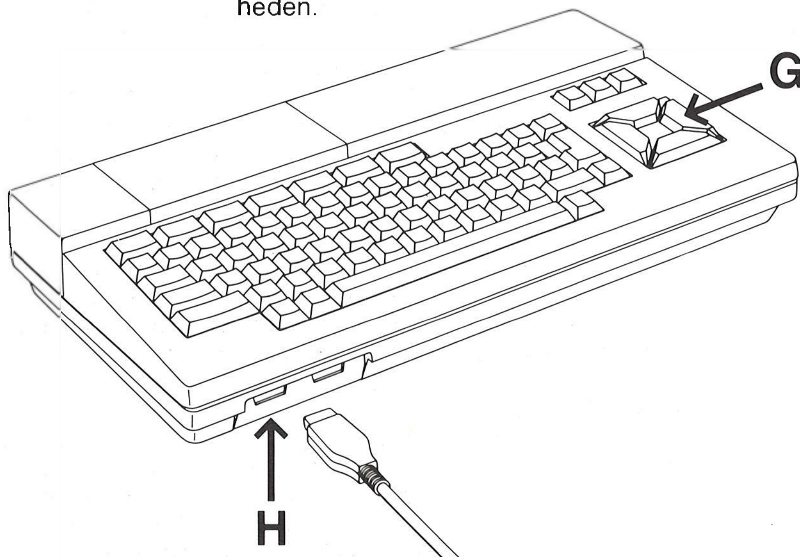


Om een diskettestation op uw MSX-computer te kunnen aansluiten hebt u een speciale interface-module nodig.

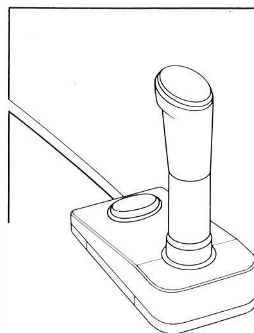


1.7 Het gebruik van spelregelaars

De meeste MSX-videospelen kunnen zowel met de cursortoetsen (G) van de computer als met spelregelaars worden gespeeld. Geeft u de voorkeur aan spelregelaars, dan kunt u deze aansluiten op de twee aansluitpunten (H) aan de voorkant van uw computer. Dezelfde aansluitingen kunnen worden gebruikt voor andere bedieningskastjes, bij voorbeeld een grafisch tableau, "game paddles" (een ander type spelregelaar) of een "muis". Vraag uw handelaar naar de mogelijkheden.



Aan de voorkant van de computer bevinden zich twee aansluitingen (H) voor spelregelaars. Merk op dat de aansluitingen de nummers 1 en 2 hebben.



1.8 Het gebruik van een afdrukeenheid

Een afdrukeenheid of printer is, naast een TV-toestel of monitor en een datarecorder, een waardevolle aanvulling van uw thuiscomputersysteem.

1.8.1 Waarom een printer ?

Er zijn twee redenen waarom een printer vrijwel onmisbaar is.
Als u een programma hebt geschreven en u wilt dit in zijn geheel bekijken, dan is het bijzonder handig als u het zwart op wit op papier hebt afgedrukt. Als u namelijk het programma met functietoets F4 op het scherm zet, dan ziet u maar 24 regels tegelijk. Als u er een gewoonte van maakt uw programma's af te drukken, nadat u ze op cassette of diskette hebt weggeschreven en voordat u het geheugen wist, dan hebt u altijd een compleet overzicht van die programma's. Dat kan later een hoop problemen en ergenis voorkomen.

De tweede reden is dat u een printer nodig hebt als u uw computer wilt gebruiken voor tekstverwerking of administratie omdat u dan op de een of andere manier het resultaat van uw werk op papier zult moeten afdrukken.

1.8.2 Welke printer ?

Er zijn verscheidene printers beschikbaar die u met uw computer kunt gebruiken. Houd er echter wel rekening mee dat de specificaties van de afdrukeenheid moeten overeenstemmen met die van de MSX-computer. U kunt problemen en teleurstelling voorkomen door een speciale MSX-printer te kiezen. Daarbij hebt u keus uit drie basistypen:

1 **Een matrixprinter**. Dit type heeft een schrijfmechanisme met een aantal fijne naalden. Elk teken wordt opgebouwd uit een aantal puntjes (een matrix). Voordelen van matrixprinters zijn een grote afdruksnelheid en de mogelijkheid in principe elk gewenst teken af te drukken.
Er zijn matrixprinters met maximaal 40 of 80 tekens per regel. Ze zijn vooral praktisch voor het afdrukken van programma's en administratieve gegevens.

2 **Een letterwielprinter**. Bij dit type zijn de (meestal 96) tekens aangebracht op een letterwiel of -schijf (ook

wel margrietwiel of "daisy wheel" geheten). De tekens worden in één keer op het papier afgedrukt. Een voordeel van dit type printer is de uitstekende afdrukkwaliteit. Om die reden worden ze vooral gebruikt voor correspondentie, teksten die in offset moeten worden gedrukt en dergelijke. Nadelen zijn dat de afdruksnelheid kleiner is dan die van een matrixprinter en dat het aantal tekens dat kan worden afgedrukt beperkt is tot wat er op de schijf staat. Er zijn echter schijven met verschillende lettertypen en speciale tekens verkrijgbaar.

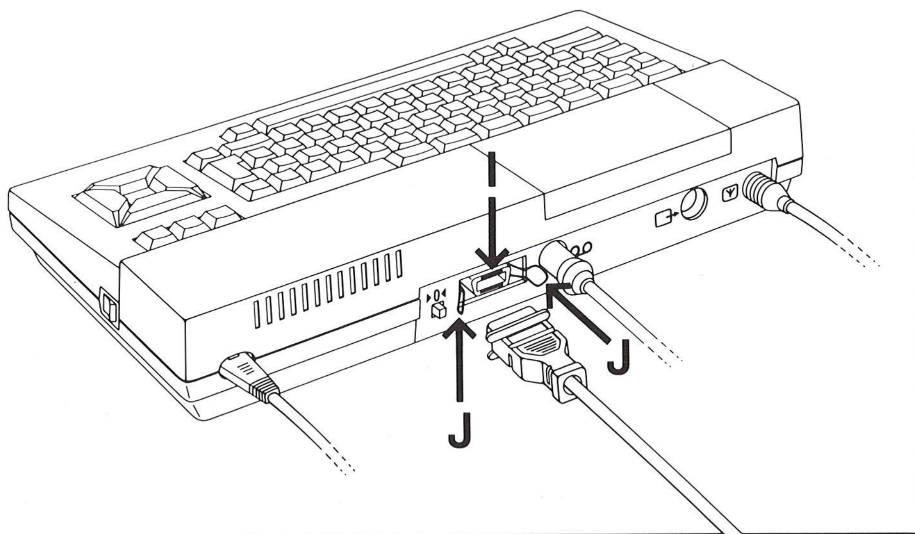
3 **Een plotter**. Dit is een zogenaamde X/Y-recorder die de tekens met een pennetje op het papier schrijft. Een voordeel van een plotter is dat u er ook uitstekend afbeeldingen, grafieken en dergelijke mee kunt tekenen, zelfs in verschillende kleuren. Een nadeel is de traagheid.

Vraag uw handelaar om een demonstratie van de verschillende machines voordat u uw keus maakt.

1.8.3 Het aansluiten van een printer

Uw MSX-computer heeft een speciale aansluiting voor een printer aan de achterkant. Het enige dat u verder nog nodig hebt is een speciale aansluitkabel. Sluit de kabel aan zoals in de afbeelding is weergegeven. Steek hierna de netsteker van de printer in het stopcontact.

Sluit de printerkabel aan op de daarvoor bestemde aansluiting (I) aan de achterkant van de computer. Zet daarna de steker vast door de twee klembeugeltjes (J) naar binnen te draaien totdat ze vastklikken.

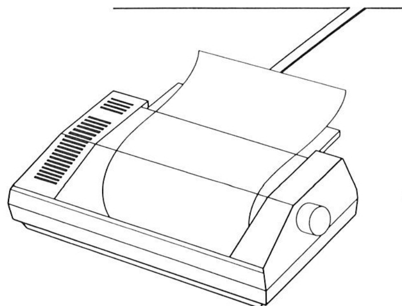


1.8.4 Werken met de printer

Eerder hebben we al gezegd dat u een programma kunt afdrukken en als "hard copy" bewaren nadat u het hebt weggeschreven op cassette of diskette en voordat u het geheugen wist. Vanzelfsprekend kunt u ook een programma van cassette of diskette inlezen in het computergeheugen en er dan alsnog een "harde afdruk" van maken.

Om te laten zien hoe dat gaat nemen we weer hetzelfde miniprogramma dat we eerder hebben gebruikt:

```
10 A$="MSX thuiscomputer   "
20 PRINT A$
30 GOTO 20
```



Zorg ervoor dat de printer is ingeschakeld en tik dan het volgende commando in:

```
LLIST
```

Let op de dubbele L.
Druk nu op RETURN. De printer zal de drie programmaregels netjes op het papier afdrukken.
Het commando LLIST wordt gebruikt om alle programmaregels, die zich in het computergeheugen bevinden, af te drukken op het papier.

U kunt de computer ook opdracht geven bepaalde gegevens tijdens het uitvoeren van het programma af te drukken. Tik de volgende regel in:

```
20 LPRINT A$
```

Druk dan op RETURN. Merk op dat in regel 20 PRINT nu is vervangen door LPRINT doordat de oorspronkelijke regel 20 is overschreven door de nieuwe.
Druk nu functietoets F5 in. De printer zal nu eindeloos de tekst "MSX thuiscomputer " afdrukken (zoals hij die eerst op het scherm zette).
U kunt het programma afbreken door tegelijk de toetsen CTRL en STOP in te drukken.

1.9 Toetsenbord

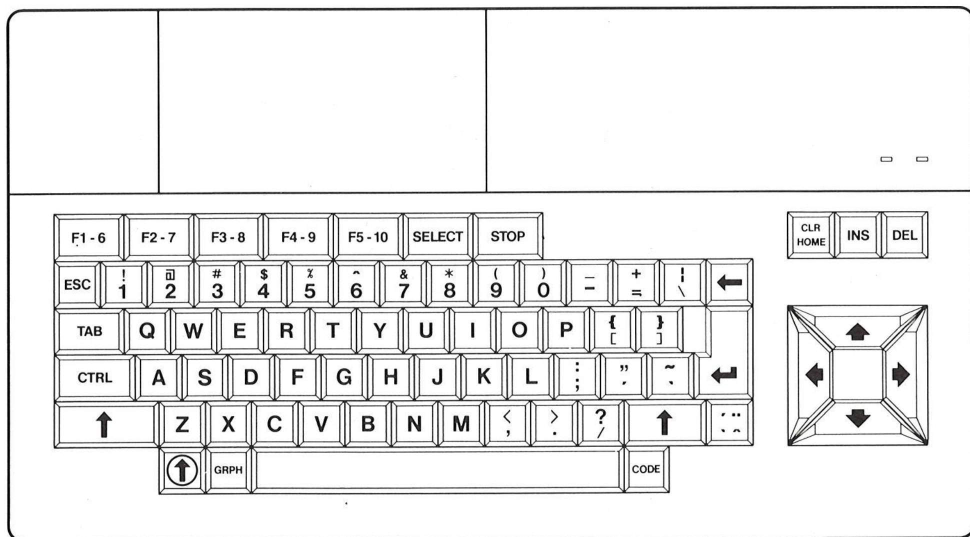
Begin gerust met alle toetsen van uw computer te proberen. Daardoor kan de computer geen schade oplopen, als u tenminste niet overdreven veel kracht zet. Om de toetsen te bedienen is trouwens maar erg weinig kracht nodig. U zult merken hoe eenvoudig het is een letter of teken op het scherm te krijgen.
Houdt u een toets langer ingedrukt dan nodig is, dan zal hij na enige tijd gaan repeteren. Er verschijnt een rij gelijke tekens op het scherm, totdat u de toets loslaat.

Het toetsenbord bestaat uit vier groepen toetsen:
1 Linksboven vijf functietoetsen, de SELECT-toets en de STOP-toets.
2 Rechtsboven drie toetsen met de opschriften CLR/HOME, INS en DEL.

3 Daaronder vier "cursor"-besturingstoetsen (de "cursor" of positieaanwijzer is het witte blokje op het scherm, dat aangeeft waar het volgende teken zal verschijnen).
4 In het midden de schrijfmachinetoetsen plus enkele speciale toetsen, gemerkt met ESC, TAB, CTRL, GRPH en CODE.

1.9.1 Functietoetsen

De functies, die u met deze toetsen kunt oproepen, kunt u zelf programmeren (zie § 2.1). Bij het inschakelen van de computer zijn deze toetsen echter voorgeprogrammeerd. De standaardfuncties van deze toetsen vindt u onderaan op het scherm.



Drukt u de hoofdlettertoets (SHIFT) in, dan worden onderaan het scherm de voorgeprogrammeerde functies F6 tot en met F10 aangegeven. Deze kunt u oproepen door tegelijkertijd de hoofdlettertoets (SHIFT) en een van de functietoetsen in te drukken. Drukt u bij voorbeeld de hoofdlettertoets en F1 tegelijk in, dan wordt functie F6 uitgevoerd.

De twee overige functietoetsen hebben de volgende betekenis (een plusteken, zoals bij CTRL + STOP, betekent steeds dat de toetsen tegelijk moeten worden ingedrukt):

SELECT — Deze toets functioneert alleen als u een programmamodule in één van de twee sleuven hebt aangebracht. Hij dient voor het kiezen van een spelvariant (SELECT betekent kies). Bij MSX-BASIC wordt deze toets niet gebruikt.

STOP — Deze toets wordt gebruikt om een programma te onderbreken. Drukt u nogmaals op de STOP-toets, dan wordt het programma hervat. De STOP-toets doet dus dienst als pauzetoets.

CTRL + STOP — Door deze twee toetsen tegelijk in te drukken wordt de uitvoering van het programma afgebroken en keert de computer terug naar de directe stand (zie § 2.1.1). U kunt de uitvoering laten hervatten door CONT in te tikken, gevolgd door RETURN.

1.9.2 Commandotoetsen

De commandotoetsen hebben de volgende functies:

CLR/HOME — De cursor gaat naar de eerste positie van de eerste regel op het scherm.

SHIFT + CLR/HOME — Drukt u deze twee toetsen tegelijkertijd in, dan wordt het scherm schoongeveegd. (de twee SHIFT-toetsen zijn gemerkt met ↑.)

INS — Na het indrukken van deze toets kunt u tekens invoegen in een bestaande regel (INS betekent Insert = invoegen).

DEL — Het indrukken van deze toets heeft tot gevolg dat het teken onder de cursor wordt gewist (DEL betekent Delete = wissen).

1.9.3 Cursor-besturingstoetsen

De pijltjes op deze toetsen geven de richting aan waarin de cursor beweegt als u de desbetreffende toets indrukt.

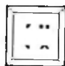
1.9.4 Schrijfmachinetoetsen

De meeste toetsen komen overeen met die van een normale schrijfmachine. Drukt u de A in, dan verschijnt de letter "a" op het scherm. Drukt u dezelfde toets tegelijk in met de hoofdlettertoets (SHIFT + A), dan komt de hoofdletter "A" op het scherm.
Links en rechts van het grote toetsenbord bevinden zich enkele speciale toetsen. Deze hebben de volgende functies:

ESC	De functie van deze toets wordt bepaald door het programma. Bij MSX-BASIC wordt deze toets niet gebruikt (ESC betekent Escape = ontsnappen).
TAB	Met deze toets kunt u de cursor naar de eerstvolgende tabulatorstop verplaatsen (TAB betekent Tabulator).
CTRL	Deze toets functioneert alleen in combinatie met een andere toets. Zie Appendix D. (CTRL betekent Control = besturen)
↑	(SHIFT) Hoofdlettertoets.
Ⓘ	(CAPS) Deze toets, gemerkt met een pijl in een cirkel, is de hoofdletter-vastzettoets. Als u hem indrukt licht de CAPS-indicator op en verschijnen er hoofdletters op het scherm als u een lettertoets indrukt (CAPS betekent Capitals = hoofdletters). Een verschil met een normale hoofdletter-vastzettoets van een schrijfmachine is dat hij alleen voor de lettertoetsen geldt. De toetsen voor cijfers en leestekens komen

niet in de hoofdletterstand. Tekens zoals $ en % kunt u alleen gebruiken als u ook nog de SHIFT-toets indrukt. De toets CAPS kan worden gebruikt om bij voorbeeld BASIC-programma's in hoofdletters in te toetsen.

←	(BS) Met deze toets kunt u het teken links van de cursor wissen. (BS betekent Backspace = spatie terug).
(RETURN)	Als u deze grote toets, gemerkt met een gebroken pijl, indrukt gaat de cursor naar het begin van de volgende regel (RETURN betekent ga terug). Deze toets moet altijd worden gebruikt om het invoeren van een BASIC-programmaregel af te sluiten, maar ook na het invoeren van gegevens waar de computer om heeft gevraagd (om te bevestigen dat de ingevoerde gegevens correct zijn). Bij andere computers heet deze toets vaak de ENTER-toets.
GRPH	Na het indrukken van deze toets kunt u met bijna alle toetsen een grafisch symbool invoeren. Drukt u tegelijkertijd SHIFT in, dan kunt u een andere reeks grafische symbolen intoetsen. (GRAPH betekent Graphics = grafische symbolen).
CODE	Bijna alle toetsen geven nu bijzondere letters, die in sommige talen voorkomen. Drukt u tegelijkertijd SHIFT in, dan kunt u een andere reeks bijzondere letters intoetsen. Door nogmaals CODE in te drukken krijgen de toetsen weer hun oorspronkelijke betekenis.



Dit is de accenttoets. U kunt er een Frans accentteken ` ^ of een trema mee op een letter zetten (alleen op klinkers).
Net als bij een schrijfmachine dient u altijd eerst de accenttoets in te drukken. Er verschijnt dan nog geen accentteken op het scherm. Druk daarna de toets met de gewenste letter in, bij voorbeeld de e. Dan zal deze letter compleet met het gekozen accentteken op het scherm verschijnen.
De vier accenttekens kunt u als volgt oproepen:
Accent-grave (è): accenttoets, gevolgd door de lettertoets.
Accent-aigu (é): SHIFT + accenttoets, gevolgd door de lettertoets.
Accent-circonflexe (ê): CODE + accenttoets, gevolgd door de lettertoets.
Trema of Umlaut (ë, ü): CODE + SHIFT + accenttoets, gevolgd door de lettertoets.

Samenvatting

Met de meeste toetsen van uw MSX-computer kunt u zes verschillende soorten tekens invoeren:

1 Kleine letters
2 Hoofdletters (SHIFT ingedrukt houden)
3 Grafische symbolen 1 (GRPH ingedrukt houden)
4 Grafische symbolen 2 (GRPH + SHIFT ingedrukt houden)
5 Taalgebonden tekens en symbolen 1 (CODE ingedrukt houden)
6 Taalgebonden tekens en symbolen 2 (CODE + SHIFT ingedrukt houden)

Een volledig overzicht van alle tekens vindt u in Appendix G.

2 MSX-BASIC

2.1 Werken met MSX-BASIC

Als u uw computer inschakelt, nadat u hem op de juiste manier hebt aangesloten, zal de volgende tekst op het scherm verschijnen:

```
MSX system
enz.
```

De computer zal nu controleren of er een programmamodule aanwezig is in één van de twee sleuven. Is dat niet het geval, dan verschijnt automatisch de volgende tekst op het scherm:

```
MSX BASIC
enz.
```

Het kleine blokje, direct onder de letters "Ok", is de positieaanwijzer of "cursor". Deze geeft aan waar het volgende teken op het scherm zal verschijnen.
Het getal geeft het aantal vrije geheugenposities aan, dat beschikbaar is voor het programma en de gegevens.
De letters "Ok" betekenen dat de computer in de zogenaamde "directe stand" staat. Dit wil zeggen dat u opdrachten kunt intoetsen, die dan onmiddellijk worden uitgevoerd.
Op dit punt kunt u ook kiezen voor de "programmastand", waarbij de instructies die u intoetst pas later worden uitgevoerd.

2.1.1 Directe stand

In de directe stand (in het Engels "Direct Mode" of "Command Mode" geheten) worden de meeste instructies (met of zonder functies) en commando's direct door de computer uitgevoerd als u RETURN indrukt. Ze worden niet in het geheugen opgeslagen.
In het Naslagwerk voor Programmeurs vindt u een compleet overzicht van alle commando's, instructies en functies.
Toets nu het volgende voorbeeld in:

```
PRINT 5+2
```

Als u een tikfout hebt gemaakt, kunt u het teken links

naast de cursor wissen met de BS-toets. Met deze toets kunt u achtereenvolgens net zoveel tekens wissen als nodig is om de fout te herstellen. Druk op RETURN als de regel correct is.
Denk eraan dat u na elk commando en elke instructie de RETURN-toets moet indrukken. MSX-BASIC zal dan onmiddellijk de instructie uitvoeren en het antwoord op het scherm zetten:

```
 7
Ok
```

"Ok" betekent dat de computer weer in de directe stand staat en wacht op de volgende instructie.

2.1.2 Programmastand

In de programmastand, ook wel indirecte stand genoemd (Engels: "Indirect Mode"), moet aan alle instructies een regelnummer voorafgaan. De instructies worden niet onmiddellijk uitgevoerd als u op RETURN drukt, maar ze worden opgeslagen in het geheugen, in volgorde van hun regelnummers.
Toets nu het volgende voorbeeld in:

```
10 PRINT 5+2
```

Druk nu RETURN in. U ziet dat er niets gebeurt, behalve dat de cursor naar de eerste positie op de volgende regel springt. Dit betekent dat programmaregel 10 nu is opgeslagen in het computergeheugen.
Een combinatie van zulke regels met instructies noemt men een programma. Als u het woord RUN intoetst, gevolgd door het indrukken van de RETURN-toets, voert de computer de instructies uit in volgorde van de regelnummers.
In de programmastand moet elke regel beginnen met een regelnummer, dat moet liggen tussen 0 en 65529. U kunt dus een groot aantal verschillende regelnummers gebruiken. Het verdient aanbeveling geen opeenvolgende regelnummers te gebruiken, zoals 1, 2, 3 enz., maar er een goede gewoonte van te maken regelnummers over te slaan. Als u de regelnummers 10, 20, 30 enz. gebruikt, kunt u tussen twee opeenvolgende regels later nog 9 regels invoegen. Dit heeft geen invloed op

het geheugengebruik. Regelnummer 65520 vraagt niet meer geheugenruimte dan regelnummer 0.

U kunt ook MSX-BASIC de regelnummers laten kiezen, eenvoudig door eerst het commando AUTO in te toetsen, gevolgd door RETURN, voordat u met programmeren begint. U ziet dat de computer dan zelf 10 als eerste regelnummer op het scherm zet. Wordt het regelnummer gevolgd door een asterisk (dus 10*), dan bevindt zich al een programmaregel met regelnummer 10 in het geheugen van de computer. Tikt u achter dit regelnummer geen nieuwe gegevens in, dan blijft de oorspronkelijke regel behouden.
Drukt u op RETURN, dan verschijnt automatisch het volgende regelnummer (bij voorbeeld 20). Het automatisch genereren van regelnummers kunt u beëindigen door tegelijkertijd de toetsen CTRL en C in te drukken.

Een programmaregel in MSX-BASIC mag uit ten hoogste 255 tekens bestaan, inclusief regelnummer en spaties. Er mogen verscheidene instructies in een programmaregel staan, mits ze worden gescheiden door een dubbele punt (:). Probeer het volgende voorbeeld:

```
10 PRINT 8+2:PRINT 6*4
RUN
 10
 24
Ok
```

MSX-BASIC heeft beide berekeningen afzonderlijk uitgevoerd omdat ze gescheiden zijn door een dubbele punt. Merk op dat * het vermenigvuldigingsteken is.

Wilt u een programma uit het geheugen verwijderen, toets dan het commando NEW in en druk op de toets RETURN.

2.1.3 Het verschil tussen commando's en instructies

Commando's en instructies mogen beide zowel in de directe stand als in de programmastand worden gebruikt. Na het uitvoeren van een commando keert MSX-BASIC altijd terug naar de directe stand. Het volgende voorbeeld kan dit verduidelijken:

```
10 PRINT 3+1
20 LIST
30 PRINT 6-3
RUN
```

LIST is een commando, dat tot gevolg heeft dat de programmaregels op het scherm worden gezet (LIST = lijst van programmaregels). Dit commando wordt gebruikt om verbeteringen in het programma te kunnen aanbrengen.
Omdat LIST een commando is en geen instructie, zal MSX-BASIC na het uitvoeren van regel 20 terugkeren naar de directe stand en regel 30 niet uitvoeren.
Een ander voorbeeld van een commando is NEW. Het commando NEW wordt gebruikt om een eventueel programma uit het geheugen te verwijderen.

Een instructie is een opdracht die door de computer wordt uitgevoerd zonder dat hij na het uitvoeren daarvan in de directe stand terugkeert. Een voorbeeld is de instructie PRINT, waarmee u de computer opdracht geeft iets op het scherm te zetten.
In het Naslagwerk voor Programmeurs kunt u opzoeken of een opdracht (Engels: "statement") een commando, een instructie of (dat kan ook nog) een functie is.

2.1.4 Het verschil tussen instructies en functies

Een functie is een opdracht die als uitkomst een getal oplevert. Met dat getal moet iets worden gedaan. Daarom moet een functie altijd vooraf worden gegaan door een instructie. FRE(0) is een voorbeeld van een functie, die wordt gebruikt om MSX-BASIC de hoeveelheid vrije geheugenruimte te laten berekenen. Probeer het volgende voorbeeld.

```
NEW
10 FRE(0)
```

Geef nu het commando RUN. U ziet dat MSX-BASIC een foutmelding op het scherm zet: "Syntax error in 10" ("syntaxis" is de leer van de zinsbouw; er is dus een fout gemaakt bij de zinsbouw in de taal MSX-BASIC). U hebt een functie gebruikt die niet wordt voorafgegaan door een instructie. Probeer nu het volgende voorbeeld:

```
10 PRINT FRE(0)
RUN
```

De computer vertelt u nu hoeveel bytes vrije geheugenruimte u nog beschikbaar hebt.

2.1.5 Het invoeren van instructies

PRINT is een instructie die u nogal vaak zult tegenkomen. Om het u gemakkelijk te maken en tijd te sparen kunt u ook een vraagteken intikkken. MSX-BASIC vertaalt het vraagteken automatisch in de instructie PRINT. Een voorbeeld om te laten zien hoe dat werkt; toets in:

```
10 ? 2+9
```

Druk op RETURN en toets het commando LIST in. Druk dan nogmaals op RETURN. Op het scherm ziet u:

```
10 ? 2+9
LIST
10 PRINT 2+9
Ok
```

MSX-BASIC heeft het vraagteken vertaald in PRINT.

2.1.6 Het gebruik van kleine letters en hoofdletters

Tot dusver hebben we voor het intoetsen van commando's en instructies steeds hoofdletters gebruikt. U mag echter ook kleine letters gebruiken. Deze worden door MSX-BASIC automatisch omgezet in hoofdletters. Voorbeeld:

```
10 print fre(0)
list
10 PRINT FRE(0)
Ok
```

Dat u na elk commando en na elke programmaregel de RETURN-toets moet indrukken zullen we er voortaan niet steeds bij zeggen.

2.1.7 Commando's en instructies met één toets

Enkele commando's en instructies hoeft u niet letter voor letter in te toetsen, want die kunt u met MSX-BASIC door middel van één toets invoeren. Daarvoor dienen de functietoetsen F1 tot en met F10. Het was u natuurlijk al opgevallen dat op de onderste regel van het scherm vijf commando's en instructies staan. Deze vijf corresponderen met de functietoetsen F1 tot en met F5. Drukt u op SHIFT, dan verschijnen de vijf commando's of instructies die corresponderen met de functietoetsen F6 tot en met F10.
U kunt een volledig overzicht opvragen van de commando's en instructies die aan de tien functietoetsen zijn toegekend door het commando KEY LIST in te toetsen (denk om RETURN).
U kunt zelf andere commando's of instructies aan de functietoetsen toekennen. Probeer het volgende voorbeeld maar eens:

```
KEY 1,"GOSUB"
```

Van nu af aan kunt u met toets F1 de instructie GOSUB oproepen, zodat u niet telkens vijf letters hoeft te gebruiken.
De instructies GOSUB, GOTO en dergelijke kunt u alleen in een programma gebruiken, de commando's LIST, RUN, AUTO en dergelijke zult u hoofdzakelijk in de directe stand gebruiken. Meestal moet u RETURN indrukken nadat u met een functietoets een commando hebt ingevoerd (niet bij RUN).

2.1.8 Wissen van de onderste regel

Het overzicht van functietoetsen op de onderste regel kunt u wissen met de instructie KEY OFF. Het omgekeerde gebeurt met de instructie KEY ON.

2.1.9 Het gebruik van "", :, ;, . en ,

MSX-BASIC zal al uw opdrachten keurig uitvoeren, mits u ze op de juiste manier intoetst. Dat geldt ook voor de leestekens, waarvan de meeste een speciale functie hebben.

Alle tekens die tussen aanhalingstekens ("") (Engels: quotation marks) staan worden door de computer pre-

cies zo op het scherm gezet. Dit kunt u proberen met het volgende voorbeeld:

```
10 PRINT 5+2
20 PRINT "5+2"
RUN
 7
5+2
Ok
```

In regel 10 zal de computer de berekening 5 + 2 = 7 uitvoeren. In regel 20 zet hij alleen de tekst op het scherm zoals die tussen aanhalingstekens staat, dat wil zeggen 5 + 2.

De dubbele punt (:) (Engels: colon) dient om twee instructies in dezelfde programmaregel van elkaar te scheiden, zoals we al eerder hebben gezien.

De puntkomma (;) (Engels: semi-colon) kan worden gebruikt in combinatie met de PRINT-instructie. De puntkomma voorkomt dat de cursor na het uitvoeren van de PRINT-instructie onmiddellijk naar de volgende regel gaat.
De puntkomma wordt ook gebruikt in combinatie met de INPUT-instructie, eveneens om te voorkomen dat de cursor naar de volgende regel gaat.

De punt (.) (Engels: period) kunt u gebruiken in combinatie met de commando's LIST, AUTO en DELETE (= wis) als u deze commando's wilt toepassen op de laatst ingevoerde of (door het programma) uitgevoerde programmaregel. Toetst u, na het laatste programmavoorbeeld, DELETE. (let op de punt) in en daarna LIST, dan zult u zien dat regel 20 gewist is; dat was de laatste regel die door het programma is uitgevoerd.

De komma (,) (Engels: comma) wordt voor verschillende doeleinden gebruikt. Op het gebruik van de komma komen we verderop in deze gebruiksaanwijzing terug.

2.1.10 Foutmeldingen

Als MSX-BASIC een fout vindt in één van uw programmaregels, zal hij een foutmelding op het scherm zetten.

In Appendix A vindt u een compleet overzicht van de foutmeldingen.

2.2 Veranderen van een programma

Een programmaregel kan gemakkelijk worden veranderd door een nieuwe programmaregel met hetzelfde regelnummer in te toetsen.
Nieuwe regels kunnen aan het programma worden toegevoegd door ze een nog niet gebruikt regelnummer te geven. MSX-BASIC zet de regels automatisch in de goede numerieke volgorde. Denk er wel aan dat u een regelnummer moet kiezen dat de regel op de juiste plaats zet. Wilt u tussen regel 10 en regel 20 een nieuwe programmaregel invoegen, gebruik dan een regelnummer tussen 11 en 19 (bij voorkeur 15).
Programmaregels kunnen worden gewist met het commando DELETE (= wis). Een enkele programmaregel kunt u ook wissen door het regelnummer in te toetsen, onmiddellijk gevolgd door het indrukken van de RETURN-toets.
Het is ook mogelijk programmaregels op het scherm te veranderen. Men noemt dit "screen editing" (letterlijk: redigeren op het scherm).

2.2.1 "Screen editing"

"Screen editing" wil zeggen dat u programmaregels, die op het scherm staan, kunt veranderen met behulp van de toetsen BS (←), DEL en INS (zie § 1.3.2 en 1.3.4). Probeer dit met het volgende voorbeeld:

```
NEW
10 pontt
```

Druk hierna op RETURN. U hebt blijkbaar een typefout gemaakt, want u bedoelde het woord PRINT. Toets nu LIST in en druk op RETURN. Omdat MSX-BASIC commando's, instructies en functies automatisch met hoofdletters schrijft, krijgt u op het scherm:

```
10 PONTT
```

Om dit te veranderen brengt u de cursor met de cursortoetsen naar de O van PONTT. Druk dan op INS (Insert

= invoegen). Merk op dat de cursor is veranderd in een smal streepje. Dit betekent dat elk teken dat u intoetst tussen de tekens op het scherm zal worden gevoegd. Druk op de toets R en u ziet dat de R tussen de letters P en O wordt gezet.
Druk nogmaals op INS. Daarmee schakelt u de invoegfunctie uit. De cursor verandert weer in een blokje. Dit betekent dat elk teken, dat u nu intoetst, in de plaats komt van het teken onder de cursor. Druk op de toets I en er staat:

```
10 PRINTT
```

De cursor staat nu op de N. Het enige dat we nog moeten doen is één T verwijderen. Verplaats daarom de cursor naar de eerste T en druk op DEL (Delete = wis). De eerste T is nu verdwenen en de tweede T is één plaats naar links opgeschoven. Druk nu op RETURN. MSX-BASIC brengt de gemaakte veranderingen pas aan als u op RETURN drukt.
Toetst u het commando LIST in (gevolgd door RETURN), dan ziet u dat de regel is veranderd in:

```
10 PRINT
```

En dat is precies wat u bedoelde.

2.2.2 Voorkomen van foutmeldingen

Zorg er altijd voor dat na "screen editing" nieuwe commando's en instructies op een lege regel worden ingetoetst. Laten we aannemen dat u het volgende programma op het scherm hebt staan:

```
10 PRINT 9+2
20 PRINT 3-6
```

Nemen we vervolgens aan dat de cursor op de eerste positie van de tweede regel staat (op de 2 dus) doordat u zojuist een verandering hebt aangebracht in regel 10. Toetst u nu het commando LIST in, dan is dit het resultaat:

```
10 PRINT 9+2
LISTRINT 3-6
```

Drukt u nu RETURN in, dan geeft MSX-BASIC een foutmelding omdat hij het commando LISTRINT 3-6 niet kent. U kunt dit voorkomen door de cursor naar een lege regel te verplaatsen of door tegelijkertijd de toetsen SHIFT en HOME in te drukken, voordat u LIST intoetst (SHIFT + HOME = CLS = CLear Screen = wis het scherm).

2.2.3 Wissen van de rest van een regel

Als het deel van een programmaregel vanaf de plaats waar de cursor staat tot aan het einde van die regel mag worden gewist, dan kunt tegelijkertijd CTRL en E indrukken. Als regel 20 in het laatste voorbeeld weg mag, kunt u dus ook na LIST tegelijk de toetsen CTRL en E indrukken.

2.2.4 De STOP-toets

De STOP-toets heeft geen betekenis als de computer in de directe stand staat. Voert hij echter een programma uit, dan kunt u de programma-uitvoering onderbreken door STOP in te drukken. Dan zal de cursor weer op het scherm verschijnen. Drukt u nogmaals STOP in, dan wordt de programma-uitvoering hervat.
Wilt u een programma onderbreken en terugkeren naar de directe stand, druk dan tegelijkertijd CTRL en STOP in. U krijgt dan de mededeling op het scherm: "Break in nnnnn", waarin nnnnn het regelnummer is waar MSX-BASIC mee bezig was toen het programma werd onderbroken.

2.3 Constanten en variabelen

Het geheugen van uw MSX-computer kan meer dan alleen regels met commando's, instructies en functies opslaan. Er zijn nog meer elementen die een rol spelen bij het uitvoeren van een programma. Deze kunnen worden verdeeld in twee soorten: "constanten" en "variabelen".

2.3.1 Constanten

Constanten bevatten informatie die tijdens het uitvoeren van het programma niet verandert. Er zijn twee soorten constanten: alfanumerieke en numerieke constanten.
Een alfanumerieke constante, ook wel string-constante

genoemd, kan bestaan uit ten hoogste 255 letters, cijfers en andere tekens. Alfanumerieke constanten staan altijd tussen aanhalingstekens (".... "). Voorbeelden: "Philips" en "2000". Hoewel het laatste voorbeeld uitsluitend uit cijfers bestaat, is het geen getal dat voor berekeningen kan worden gebruikt.
Een numerieke constante is een positief of negatief getal. In plaats van de decimale komma moet bij MSX-BASIC altijd de decimale punt worden gebruikt, zoals in Engels sprekende landen gebruikelijk is. Schrijf dus niet 123,45, maar: 123.45.

MSX-BASIC kent zes soorten numerieke constanten:

1 **Gehele getallen of integers**
Integers zijn gehele getallen waarvan de waarde ligt tussen -32768 en +32767. Voorbeelden: 62 en -25.

2 **Hexadecimale constanten**
Hexadecimale constanten zijn gehele getallen (integers), uitgedrukt in het zestientallig stelsel. Ze worden voorafgegaan door &H. De waarde ligt tussen &H0000 en &HFFFF.
Voorbeeld: &H3E (= 62 in de decimale notatie).

3 **Octale constanten**
Octale constanten zijn gehele getallen (integers), uitgedrukt in het achttallig stelsel. Ze worden voorafgegaan door &O. De waarde ligt tussen &O000000 en &O177777. Voorbeeld: &O76 (= 62 decimaal).

4 **Binaire constanten**
Binaire constanten zijn gehele getallen (integers), uitgedrukt in het tweetallig stelsel. Ze worden voorafgegaan door &B. De waarde ligt tussen &B0000000000000000 en &B1111111111111111.
Voorbeeld: &B111110 (= 62 decimaal).

5 **Enkele-precisieconstanten**
Constanten met enkele precisie kunnen een waarde hebben tussen 1E-64 en 9E+62. Het hoeven geen gehele getallen (integers) te zijn. Getallen van meer dan 14 cijfers worden door MSX-BASIC automatisch in de wetenschappelijke notatie door MSX-BASIC automatisch in de wetenschappelijke notatie geschreven, dat wil zeggen met de letter E voor exponent.
Voorbeelden: 15.73, -178.3 en 2.34E+8
(= 2.34×10^8 = 234 000 000, of in computertaal 2.34*10^8).

Van constanten met enkele precisie zijn alleen de eerste zes cijfers nauwkeurig.

6 **Dubbele-precisieconstanten**
Constanten met dubbele precisie kunnen eveneens een waarde hebben tussen 1E-64 en 9E+62. Het verschil met enkele-precisieconstanten is dat de eerste veertien cijfers nauwkeurig zijn.

Opmerking:
Hexadecimale, octale en binaire constanten zijn niets anders dan integers, op een andere manier geschreven. Soms heeft dat voordelen. &H3E, &O76 en &B111110 zijn dus drie alternatieve manieren om het decimale getal 62 te schrijven. MSX-BASIC kan dus eigenlijk met drie soorten constanten rekenen: integers, enkele-precisieconstanten en dubbele-precisieconstanten.

2.3.2 Variabelen

Een variabele is een grootheid waarvan de waarde kan veranderen tijdens het uitvoeren van het programma. Elke variabele krijgt een naam, die de computer in staat stelt de waarde van de variabele te vinden. Er zijn twee soorten variabelen: alfanumerieke variabelen, ook wel stringvariabelen geheten, en numerieke variabelen.

2.3.3 Stringvariabelen

De naam van stringvariabelen moet beginnen met een letter en eindigen met het dollarteken $. De naam mag een willekeurige lengte hebben, maar MSX-BASIC herkent alleen de eerste twee tekens. Voorbeelden van toegestane namen: A$ = "Philips"; AB$ = "2000"; P1$ = "MSX"; P2$ = "BASIC"; NAAM$ = "Rob". Zie ook § 2.3.7.

Opmerking 1: A$ spreekt men uit als A-string.
Opmerking 2: MSX-BASIC maakt bij namen geen onderscheid tussen kleine en hoofdletters; hij ziet a$ en A$ dus als dezelfde stringvariabele.

2.3.4 Numerieke variabelen

Een numerieke variabele kan een positieve of een negatieve waarde hebben. De naam van een numerieke variabele moet met een letter beginnen en mag een willekeurige lengte hebben, maar MSX-BASIC herkent

alleen de eerste twee tekens. Voorbeelden van toegestane namen: A=23.4; AB=-25; P1=0.001.

2.3.5 Integers; enkele en dubbele precisie

Bij numerieke constanten en variabelen moeten we onderscheid maken tussen gehele getallen (integers), getallen met enkele precisie en getallen met dubbele precisie.

Integers zijn gehele getallen met een waarde tussen -32768 en +32767. Deze getallen nemen twee geheugenplaatsen in beslag. Om aan te geven dat het om integers gaat moeten integer-constanten en de namen van integer-variabelen eindigen met het procentteken %. Voorbeelden: 12.5%; A%=10.

Enkele-precisiegetallen nemen vier geheugenplaatsen in. Van de waarde van de variabele of de constante zijn alleen de eerste zes cijfers nauwkeurig. Enkele-precisieconstanten en de namen van enkele-precisievariabelen moeten eindigen met een uitroepteken (!) om aan te geven dat het om enkele-precisiegetallen gaat. Voorbeelden: 881.917!; A!=123.456.
Enkele-precisiegetallen met meer dan 14 cijfers worden automatisch weergegeven in de wetenschappelijke notatie. Ze gebruiken de letter E als symbool voor exponent. Voorbeeld: 2.34745E+8.

Dubbele-precisiegetallen nemen acht geheugenplaatsen in. Van de waarde van de variabele of de constante zijn de eerste veertien cijfers nauwkeurig. Dubbele-precisieconstanten en de namen van dubbele-precisievariabelen mogen eindigen met een "hekje" # om aan te geven dat het om dubbele-precisiegetallen gaat. Het hekje mag worden weggelaten omdat MSX-BASIC automatisch in dubbele precisie rekent, tenzij anders is aangegeven. Voorbeelden: 23.33#; A# = 123.45678901234.
Dubbele-precisiegetallen met meer dan 14 cijfers worden automatisch in de wetenschappelijke notatie weergegeven. Voorbeeld: 3.7685297133467E-7.

Als een variabelennaam niet wordt gevolgd door $, %, ! of #, neemt MSX-BASIC aan dat het om een variabele

met dubbele precisie gaat en zullen acht geheugenposities worden gereserveerd.

Voorbeelden:

ME#	Dubbele-precisievariabele; vraagt 8 geheugenplaatsen.
ME	Dubbele-precisievariabele; vraagt 8 geheugenplaatsen.
ZS!	Enkele-precisievariabele; vraagt 4 geheugenplaatsen.
PT%	Integer-variabele; vraagt 2 geheugenplaatsen.
AB$	Stringvariabele; vraagt 3 geheugenplaatsen plus evenveel plaatsen als er tekens in de string zijn.

Uit deze voorbeelden blijkt dat u geheugenruimte kunt besparen door ! of % toe te voegen aan de naam van numerieke variabelen als de nauwkeurigheid niet groter hoeft te zijn dan 6 cijfers (!) of als de waarde tussen -32768 en +32767 ligt (%). Bovendien werkt uw programma dan sneller omdat MSX-BASIC met kleinere getallen kan rekenen.
In het Engels heet het toevoegen deze speciale tekens: type declaration.

2.3.6 Type-conversie

Desgewenst rekent MSX-BASIC numerieke variabelen om van het ene type in het andere.

Voorbeeld 1:

```
NEW
10 A%=13.03
20 PRINT A%
RUN
 13
```

In dit voorbeeld wordt een constante met vaste decimale punt toegekend aan een integer-variabele. De cijfers achter de decimale punt worden weggelaten.

Voorbeeld 2:

```
NEW
10 E#=6!/7!
```

```
20 PRINT E#
RUN
 .85714285714286
```

Hoewel de berekening 6!/7! in enkele precisie wordt uitgevoerd, is de uitkomst weergegeven in dubbele precisie omdat de variabele E# in dubbele precisie is aangegeven. In dit voorbeeld zijn de laatste cijfers echter niet nauwkeurig omdat de berekening in enkele precisie is uitgevoerd.

Voorbeeld 3:

```
NEW
10 E!=6/7
20 PRINT E!
RUN
 .857143
```

Hier wordt de berekening in dubbele precisie uitgevoerd, maar de uitkomst is weergegeven in enkele precisie omdat de variabele E! in enkele precisie is aangegeven.

Voorbeeld 4:

```
NEW
10 B!=SQR(2)
20 A=B!
30 PRINT B!;A
RUN
 1.41421  1.41421
```

Hoewel A een dubbele-precisievariabele is, wordt de uitkomst in zes cijfers weergegeven. Dat komt doordat de dubbele-precisievariabele A gelijk is gemaakt aan de enkele-precisievariabele B!.

Het omzetten (converteren) van typen en variabelen kan ook gebeuren met de functies CDBL, CINT, CSNG en STR$. Zie voor deze functies het Naslagwerk voor Programmeurs.

2.3.7 Namen van variabelen

U bent vrij in het kiezen van namen voor numerieke en stringvariabelen, mits u de volgende regels in acht neemt:

1 De naam moet altijd beginnen met een letter (A...Z).
2 De namen mogen zo lang zijn als u wilt (al kost elk teken een geheugenplaats), maar MSX-BASIC kijkt alleen naar de eerste twee tekens.
3 Namen die gereserveerd zijn voor MSX-BASIC mogen niet worden gebruikt. Deze woorden staan in Appendix F.
4 MSX-BASIC maakt geen onderscheid tussen kleine letters en hoofdletters.

Niet toegestaan zijn dus: 1A% (begint met een cijfer), ALFA$ en ALLE$ in hetzelfde programma (beginnen beide met AL), ab$ en AB$ (MSX-BASIC maakt geen onderscheid tussen kleine letters en hoofdletters), TO, PRINT$, LIST! enz. (TO, PRINT en LIST zijn "gereserveerde" woorden).

2.3.8 Het toekennen van waarden aan variabelen

Aan een variabele kan een waarde of een "inhoud" worden toegekend met de instructie LET.
Voorbeeld:

```
10 LET A=10
20 LET B$="Hallo"
```

Het woord LET is niet beslist noodzakelijk. Het effect is hetzelfde als u dit woord weglaat. Dat zullen we voortaan dan ook doen.
Een andere manier om aan een variabele een waarde of inhoud toe te kennen is met de instructie INPUT. Bij het uitvoeren van deze instructie wacht de computer totdat de gevraagde informatie is ingetoetst via het toetsenbord. Als RETURN wordt ingedrukt, wordt de ingetoetste informatie aan de variabele toegekend. Om aan de gebruiker te laten weten welke soort informatie hij moet intoetsen, kan aan de INPUT-instructie een tekst worden toegevoegd. Voorbeeld:

```
NEW
10 INPUT "Voer een getal in";A
```

```
20 INPUT "Wat is uw naam";B$
RUN
Voer een getal in ?
```

Het vraagteken wordt door MSX-BASIC automatisch aan de tekst toegevoegd. In regel 10 accepteert de computer alleen numerieke informatie. In regel 20 worden letters, cijfers en andere tekens (met uitzondering van komma's) geaccepteerd.

2.3.9 Enkele bijzondere variabelen

Met MSX-BASIC hebt u de beschikking over enkele speciale variabelen, die u kunt gebruiken in uw programma's. Dat zijn: TIME, VDP, BASE en SPRITE$. Voorbeeld:

```
NEW
10 PRINT "Start":TIME=0
20 FOR I%=1 TO 8000:NEXT I%
30 PRINT "Het is nu";TIME/50;"seco
nden later"
RUN
Start
Het is nu 6.46 seconden later
```

Dit voorbeeld laat zien dat de computer bijna 4 seconden nodig heeft om de FOR...NEXT-lus in regel 20 te doorlopen.

De speciale variabele SPRITE$ is uitvoerig beschreven in § 2.12. De variabelen VDP en BASE zijn alleen goed te gebruiken door degenen die goed bekend zijn met de werking van de videogenerator.

2.4 Berekeningen

2.4.1 Volgorde van bewerkingen

MSX-BASIC kan vijf rekenkundige bewerkingen direct uitvoeren:

1 Optellen (+)
2 Aftrekken (-)
3 Vermenigvuldigen (*)
4 Delen (/)
5 Machtsverheffen (^)

Deze berekeningen kunnen in één instructie worden gecombineerd, bij voorbeeld:

```
NEW
10 PRINT 40/5+3
RUN
 11
```

Denk aan de regel "Meneer Van Dale Wacht Op Antwoord", die de volgorde aangeeft waarin de bewerkingen worden uitgevoerd:

1 Machtsverheffen
2 Vermenigvuldigen
3 Delen
4 (Worteltrekken)
5 Optellen
6 Aftrekken

De computer houdt ook deze volgorde aan, zij het dat worteltrekken niet met één enkel teken kan gebeuren. Natuurlijk kan de volgorde van de bewerkingen worden veranderd door haakjes te gebruiken, bij voorbeeld:

```
NEW
10 PRINT 40/(5+3)
RUN
 5
```

2.4.2 Integer-deling

Een deling waarvan de uitkomst een geheel getal is, waarbij dus de cijfers achter de komma (= decimale punt) worden weggelaten, noemt men integer-deling. Een integer-deling kan worden uitgevoerd met behulp van de breukstreep die de verkeerde kant uit wijst, in het Engels "reverse slash" of "back slash" geheten. Probeer eerst de gewone deling en daarna de integer-deling, en let op het verschil:

```
NEW
10 PRINT 10/4
20 PRINT 10\4
RUN
 2.5
 2
```

De uitkomst van de integer-deling is een geheel getal. De cijfers achter de komma zijn weggelaten. Deler en deeltal moeten tussen -32768 en +32767 liggen, maar hoeven geen gehele getallen te zijn (10.1 en 4.3 in plaats van 10 en 4 geven hetzelfde resultaat).

2.4.3 Modulus-deling

Een modulus-deling is in zekere zin het complement van een integer-deling: hij deelt de deler zo vaak mogelijk op het deeltal en geeft wat er overblijft (de "rest") als resultaat. Een modulus-deling kan worden uitgevoerd met de functie MOD. Een voorbeeld:

```
NEW
10 PRINT 13 MOD 5
RUN
 3
```

13 : 5 = 2 met als rest 3. Nog een voorbeeld, waaruit blijkt dat deler en deeltal geen gehele getallen hoeven te zijn:

```
NEW
10 PRINT 25.68 MOD 6.99
RUN
 1
```

In dit voorbeeld worden de getallen 25.68 en 6.99 eerst omgezet in integers (25 en 6), waarna de modulus-deling wordt uitgevoerd. Het quotiënt is 4 en de rest 1.

2.4.4 Delen door nul en "overflow"

In de wiskunde is delen door nul niet mogelijk omdat het resultaat altijd oneindig groot zou zijn. Ook MSX-BASIC kan niet door nul delen. Komt hij tijdens de uitvoering van het programma een deling door nul tegen, dan volgt de foutmelding "Division by zero" (Deling door nul) en wordt de programma-uitvoering afgebroken.
Als de uitkomst van een berekening groter is dan de "ruimte" die is gereserveerd voor de variabele, volgt de foutmelding "Overflow".
Voorbeeld:

```
NEW
10 A%=123*678:PRINT A%
RUN
Overflow in 10
```

De uitkomst is 83394, en dat is groter dan +32767, de maximale waarde van de integer-variabele A%. Nogmaals dezelfde berekening:

```
NEW
10 A=123*678:PRINT A
RUN
 83394
```

Nu kan de berekening wel worden uitgevoerd doordat MSX-BASIC de variabele A (zonder toevoeging) als dubbele-precisievariabele beschouwt.

2.4.5 Variabelen

Als in een wiskundige bewerking een variabele wordt gebruikt, dan rekent MSX-BASIC met de waarde van die variabele. Bij voorbeeld:

```
NEW
10 A=10:B=5
20 C=A*B
30 PRINT C
RUN
 50
```

In regel 20 wordt de waarde van A (in dit voorbeeld 10) vermenigvuldigd met de waarde van B (5). Geeft u in regel 10 andere waarden aan A en B, dan wordt in regel 20 een andere waarde voor C berekend.
Met string-variabelen kunnen geen rekenkundige en wiskundige bewerkingen worden uitgevoerd. Wel is het mogelijk string-constanten te "concateneren", dat wil zeggen aaneen te rijgen. Voorbeeld:

```
NEW
10 B$="MSX ":C$="- computer"
20 A$=B$+C$
30 PRINT A$
40 PRINT B$+"- BASIC"
```

```
RUN
MSX - computer
MSX - BASIC
```

2.4.6 Wiskundige functies

Een aantal veel voorkomende wiskundige berekeningen kunnen door MSX-BASIC worden uitgevoerd.

ABS	Berekent de absolute waarde van een getal, dat wil zeggen met weglating van een eventueel minteken.
COS	Berekent de cosinus van een hoek, waarbij die hoek is uitgedrukt in radialen.
EXP	EXP(n) berekent de n-de macht van e (het grondtal van de natuurlijke logaritmen). n moet kleiner zijn dan 146.
LOG	Berekent de natuurlijke logaritme (met het grondtal e) van een getal.
SIN	Berekent de sinus van een hoek, waarbij die hoek is uitgedrukt in radialen.
SQR	Berekent de vierkantswortel uit een getal.
TAN	Berekent de tangens van een hoek, waarbij die hoek is uitgedrukt in radialen.

Andere wiskundige functies kunnen worden uitgevoerd door combinatie van de hier genoemde functies. Een overzicht hiervan vindt u in Appendix B.

U kunt ook zelf wiskundige functies definiëren met de instructie DEF FN, zoals in het volgende voorbeeld:

```
NEW
10 DEF FNAB(X,Y)=X^3/Y^2
20 X=6:Y=2
30 T=FNAB(X,Y)
40 PRINT T
RUN
 54
```

De waarden voor X en Y, die in regel 20 zijn gedefinieerd, worden in regel 30 ingevuld in de wiskundige definitie uit regel 10. De variabele T krijgt dus de waarde die wordt bepaald door de functie $6^3/2^2$.

2.5 MSX-BASIC en de datarecorder

2.5.1 Inleiding

Op uw MSX-computer kan een datarecorder worden aangesloten, waarmee programma's en bestanden ("files") op een cassette kunnen worden weggeschreven en later weer kunnen worden ingelezen.

2.5.2 Het wegschrijven van programma's

Een programma kan op cassette worden weggeschreven met het commando CSAVE. De procedure is als volgt:

1 Plaats een lege casstte in de recorder en spoel die geheel terug.
2 Spoel de cassette een stukje vooruit totdat de aanloopstrook de kop is gepasseerd (tenzij u een speciale datacassette zonder aanloopstrook gebruikt).
3 Druk gelijktijdig de afspeel- en de opneemtoets (PLAY en REC) in.
4 Toets het commando CSAVE in, onmiddellijk gevolgd door de naam van het programma dat u wilt wegschrijven. Zet de naam tussen aanhalingstekens, bij voorbeeld: CSAVE"Naam van programma".
5 Druk op RETURN.
6 Wacht op Ok

Een programma of bestand dat eerder op dezelfde cassette is weggeschreven wordt zonder waarschuwing overschreven. Hou dus nauwgezet aantekening van de programma's en bestanden die u hebt weggeschreven, met de tellerstanden erbij. U kunt ook via de luidspreker van de cassetterecorder de band afluisteren. Opgenomen bestanden en programma's zijn herkenbaar aan het geluid als van een cirkelzaag.

2.5.3 Het inlezen van programma's

Een programma kan vanaf de cassette worden teruggelezen in het geheugen van de computer met behulp van het commando CLOAD:

1 Plaats de cassette met het programma in de recorder en spoel hem geheel terug.
2 Druk de afspeeltoets (PLAY) van de recorder in.
3 Toets het volgende commando in: CLOAD"Naam van programma".

4 Druk op RETURN.
5 Wacht op Ok
Als CLOAD niet wordt gevolgd door een naam, zal het eerste programma worden ingelezen dat op de cassette staat. Vindt MSX-BASIC een programma met een andere naam dan u hebt ingetoetst, dan verschijnt de mededeling "SKIP (PROGRAMMANAAM)" op het scherm. MSX-BASIC gaat verder met zoeken tot het juiste programma is gevonden. Dan verschijnt de mededeling "FOUND (PROGRAMMANAAM)" op het scherm en wordt het programma ingelezen in het computergeheugen.
Als dat is gebeurd, keert MSX-BASIC terug naar de directe stand.

2.5.4 Het wegschrijven van gegevensbestanden

Gegevens die tijdens de uitvoering van een programma zijn verzameld, kunnen eveneens op cassette worden weggeschreven. Deze gegevens worden sequentieel, dat wil zeggen achter elkaar weggeschreven. Ze vormen te zamen een gegevensbestand, vaak aangeduid met de Engelse benaming "file". Elk bestand op cassette heeft een eigen naam en moet worden "geopend" voordat er gegevens naar het bestand kunnen worden weggeschreven. Het aantal bestanden dat een programma gebruikt moet tevoren worden gespecificeerd met de instructie MAXFILES. Zie het volgende voorbeeld:

```
5 MAXFILES=1
10 OPEN "CAS:NUM" FOR OUTPUT AS #1
20 A=15:B=23.5:C=10:D=25.2:E=13
30 PRINT #1,A;B;C
40 PRINT #1,D;E
50 CLOSE #1
```

In regel 10 wordt een bestand met de naam "NUM" geopend. Het gaat om een numeriek bestand, maar u mag natuurlijk in plaats van "NUM" een andere naam kiezen. Het woord CAS betekent dat het bestand zal worden weggeschreven op cassette. Het eigenlijke wegschrijven op cassette gebeurt met de instructie PRINT #. Na elke PRINT #-instructie zet MSX-BASIC CR (Carriage Return) en LF (Line Feed) op de band. Deze twee termen komen uit de schrijfmachinewereld. "Car-

riage Return" is "wagen terug" en "Line Feed" is één regel omlaag. Samen hebben deze twee instructies tot gevolg dat de schrijfmachine naar het begin van de volgende regel gaat.

Om het voorgaande aanschouwelijk te maken het volgende. Hetzelfde gebeurt bij schrijven op het scherm met de instructie PRINT. Als u een regel afsluit met RETURN genereert de computer een CR en een LF en gaat de cursor naar het begin van de volgende regel. Dit betekent dat de manier waarop de gegevens op de band worden gezet veel overeenkomst vertoont met de manier waarop ze op het scherm zouden worden gezet. Om daarvan een indruk te krijgen kunt u in het laatste programmavoorbeeld in regel 10 CAS vervangen door CRT (Cathode Ray Tube = kathodestraalbuis of beeldbuis). Na RUN komen de variabelen op het scherm in plaats van op de band.

MSX-BASIC plaatst verder automatisch een komma tussen de variabelen. Bij dit voorbeeld komen de gegevens als volgt op de band te staan:

15	,	23.5	,	10	CR	LF	25.2	,	13	CR	LF

Bij het wegschrijven van alfanumerieke variabelen (strings) moet u er zelf voor zorgen dat deze worden gescheiden door een komma, zoals in regel 30 van het volgende programma:

```
10 MAXFILES=2
20 OPEN "CAS:NAAM" FOR OUTPUT AS #1
30 OPEN "CRT:NAME" FOR OUTPUT AS #2
40 A$="JOHN":B$="PETE"
50 PRINT #1,A$","B$
60 PRINT #2,A$","B$
70 CLOSE #1,#2
```

Dit geeft het volgende resultaat op de band (en op het scherm):



Zorg ervoor dat de punten 1, 2 en 3 van § 2.5.2 zijn uitgevoerd voordat u gegevens naar de cassette wegschrijft.

2.5.5 Het inlezen van gegevensbestanden

Gegevens, weggeschreven op cassette, kunnen met de instructie INPUT# worden teruggelezen in het geheugen van de computer. Het volgende programma leest de numerieke gegevens, die we met het eerste programma hebben weggeschreven, weer in:

```
5 MAXFILES=1
10 OPEN "CAS:NUM" FOR INPUT AS #1
20 INPUT #1,V,W,X
30 INPUT #1,Y,Z
40 CLOSE #1
```

Voor het inlezen van de alfanumerieke gegevens kunt u het volgende programma gebruiken:

```
5 MAXFILES=1
10 OPEN "CAS:NAAM" FOR INPUT AS #1
20 INPUT #1,Y$,Z$
40 CLOSE #1
```

Een INPUT#-instructie kent de gegevens van de cassette tot de eerstvolgende komma of tot CR + LF toe aan de variabelen. Het aan de variabelen toekennen van de gegevens tot de eerstvolgende CR + LF gebeurt met de instructie LINE INPUT#, zoals in het volgende voorbeeld:

```
5 MAXFILES=1
10 OPEN "CAS:NAAM" FOR INPUT AS#1
20 LINE INPUT #1,X$
40 CLOSE #1
```

Dit programma kent de string JOHN,PETE, dus tot aan CR + LF, toe aan de stringvariabele X$. De oorspron-

kelijke stringconstanten "JOHN" en "PETE" zijn nu dus toegekend aan één stringvariabele X$.

Met de functie EOF (End Of File = einde van bestand) kunt u controleren of het hele bestand is ingelezen. Zorg er altijd voor dat u de punten 1 en 2 van § 2.5.3 uitvoert voordat u een gegevensbestand terugleest in het computergeheugen.

2.5.6 Programma voor een adressenbestand

Het volgende programma kunt u gebruiken om een bestand met namen, adressen en dergelijke op te bouwen, weg te schrijven op cassette en later weer in te lezen.

Opmerking:
Als een programmaregel langer is dan een regel op het scherm, gaat de cursor automatisch naar de volgende regel. Daarbij bekommert MSX-BASIC zich niet om de afbreekregels. De zetcomputer, waarmee deze tekst is gezet, doet dat echter wel. Het gevolg is dat een programma, dat u intikt op uw MSX-computer, er niet altijd precies hetzelfde uitziet als het hier is afgedrukt. De gedrukte tekst kan op een andere plaats naar de volgende regel gaan dan op het scherm van uw computer.

```
10 SCREEN 0:KEY OFF:WIDTH 39:MAXFI
LES=1
20 CLS:LOCATE 10,2:PRINT "MENU"
30 ON KEY GOSUB 100,400,700
40 LOCATE 5,5:PRINT "F1 = Adres in
voeren"
50 LOCATE 5,7:PRINT "F2 = Adres af
drukken"
60 LOCATE 5,9:PRINT "F3 = Einde pr
ogramma"
70 KEY(1) ON:KEY(2) ON:KEY(3) ON
80 GOTO 80
100 CLS
110 PRINT:PRINT "Breng uw datareco
rder in gereedheid"
115 PRINT "en druk tegelijk PLAY e
n REC in"
120 PRINT:PRINT:INPUT "Klaar (J/N)
```

```
";F$
130 IF F$="N" GOTO 120
140 IF F$="n" GOTO 120
150 OPEN "CAS:ADRES" FOR OUTPUT AS
#1
160 CLS
170 LOCATE 1,5:INPUT "Naam";A$
180 LOCATE 1,7:INPUT "Adres";B$
190 LOCATE 1,9:INPUT "Woonplaats";C$
200 LOCATE 1,11:INPUT "Telefoon";D$
210 LOCATE 1,20:INPUT "Is dit corre
ct (J/N)";F$
220 IF F$="N" GOTO 160
230 IF F$="n" GOTO 160
240 PRINT #1,A$",'B$","C$","D$
250 LOCATE 1,22:INPUT "Nog een adre
s (J/N)";F$
260 IF F$="N" GOTO 290
270 IF F$="n" GOTO 290
280 GOTO 160
290 CLOSE #1:RETURN 20
400 CLS
410 PRINT:PRINT "Breng uw datarecor
der ingereedheid"
415 PRINT "en druk de PLAY-toets in
."
420 PRINT:PRINT:INPUT "Klaar (J/N)"
;F$
430 IF F$="N" GOTO 420
440 IF F$="n" GOTO 420
450 OPEN "CAS:ADRES" FOR INPUT AS
#1
460 CLS:PRINT
470 IF EOF(1) THEN 570
480 INPUT #1,A$,B$,C$,D$
490 PRINT "Naam       :";A$:PRINT
500 PRINT "Adres      :";B$:PRINT
510 PRINT "Woonplaats:";C$:PRINT
520 PRINT "Telefoon   :";D$
530 LOCATE 1,22:INPUT "Volgende adr
es (J/N)";F$
540 IF F$="N" GOTO 570
550 IF F$="n" GOTO 570
560 GOTO 460
```

```
570 CLOSE #1:RETURN 20
700 CLS:PRINT "Einde programma"
710 END
```

Probeer dit programma door een klein aantal adressen in te voeren. Als u op functietoets F1 drukt, wordt het laatste ingevoerde adres op cassette weggeschreven. Drukt u op F2, dan wordt het volgende adres van de cassette ingelezen in het geheugen. Vergeet niet de cassette terug te spoelen voordat u probeert adressen in te lezen.

Werkt het programma niet naar behoren, controleer het dan zorgvuldig op tikfouten.

In het programma wordt een aantal commando's en instructies gebruikt die in deze gebruiksaanwijzing nog niet zijn beschreven. Zie hiervoor het Naslagwerk voor Programmeurs. Omgekeerd zijn enkele instructies, die te maken hebben met de datarecorder, hier niet beschreven. Dat geldt voor: BLOAD, BSAVE, LOAD "CAS", SAVE "CAS", MERGE en MOTOR. Zie ook hiervoor het Naslagwerk.

Lees vóór het inlezen van voorbespeelde programmacassettes eerst de gebruiksaanwijzing daarvan.

2.6 Tabellen

2.6.1 Inleiding

Als u in uw programma veel numerieke of alfanumerieke constanten gebruikt, verdient het aanbeveling deze in een lijst of tabel op te nemen. Dat kan door gebruik te maken van de instructie DATA. De constanten worden uit de tabel gelezen met de READ-instructie, zoals in het volgende programmavoorbeeld:

```
NEW
10 READ X$:PRINT X$
20 GOTO 10
DATA jan.,febr.,maart
RUN
jan.
```

```
febr.
maart
Out of DATA in 10
```

Merk op dat de strings in de DATA-regel van elkaar zijn gescheiden door een komma (dat geldt ook voor numerieke constanten). De READ-instructie in regel 10 leest de eerste constante uit de tabel. De GOTO-instructie in regel 20 maakt dat het programma terugkeert naar regel 10, waarna de tweede constante wordt gelezen enz. De foutmelding "Out of DATA in 10" wordt veroorzaakt door het feit dat na drie keer uitvoeren van de READ-instructie geen elementen meer voorradig zijn in de DATA-regel.

Met de instructie RESTORE kunt u de computer opdracht geven van voren af aan te beginnen met het lezen van de elementen van de DATA-regel. Voegt u aan het laatste programma de volgende regel toe:

```
15 RESTORE
```

dan blijkt het woord "jan." eindeloos op het scherm te worden gezet, totdat u het programma onderbreekt door gelijktijdig CTRL en STOP in te drukken.
Wis nu het geheugen met het commando NEW.

2.6.2 Tabellen met variabelen

Ook variabelen kunt u opnemen in een tabel (in het Engels vaak aangeduid met "array"). Dit kunt u doen met de instructie DIM. Een voorbeeld:

```
10 DIM A(20)
```

Dit betekent dat in het geheugen 21 "velden" worden gereserveerd (0 tot en met 20), die alle de naam "A" hebben. Het is duidelijk dat het opgeven van de naam "A", als u een variabele wilt opvragen, niet meer voldoende is omdat ze alle dezelfde naam hebben. U kunt in dit geval een bepaalde variabele opvragen door aan de naam een nummer (index) tussen haakjes toe te voegen:

```
20 PRINT A(3)
```

In plaats van een nummer kunt u ook de naam van een tweede (numerieke) variabele gebruiken:

```
NEW
10 DIM A(18)
20 A(3)=15:B=0
30 PRINT A(B)
RUN
 0
```

MSX-BASIC geeft het antwoord 0 omdat de inhoud van geheugenveld A(0) nog niet is ingevuld, en dus gelijk is aan nul. Verander regel 20 nu als volgt:

```
20 A(3)=15:B=3
```

Als u het programma nu opnieuw laat uitvoeren, geeft MSX-BASIC als antwoord "15", want dat is de inhoud van geheugenveld A(3).

2.6.3 Twee- en meerdimensionale tabellen

Het is ook mogelijk in MSX-BASIC twee-, drie- en meerdimensionale tabellen te DIMensioneren. Voor een tweedimensionale tabel reserveert u als volgt de nodige geheugenruimte:

```
10 DIM A(1,3)
```

De ruimte die in het geheugen wordt gereserveerd ziet er als volgt uit:

	0	1	2	3
0				
1				

Als u in een programmaregel aan variabele A een waarde wilt toekennen of een waarde uit de tabel wilt lezen, moeten beide dimensies van A worden genoemd. Bij voorbeeld:

```
20 A(1,2)=64
```

Deze instructie plaatst het getal 64 in regel 1, kolom 2 van de gereserveerde geheugenruimte.

	0	1	2	3
0				
1			64	

MSX-BASIC reserveert zelf 11 geheugenvelden (0 tot en met 10) bij het inschakelen. Dimensioneren met de DIM-instructie hoeft dus alleen als u meer dan 11 velden wilt reserveren.

Grote tabellen kunnen tamelijk veel geheugenruimte consumeren. Om dat ruimtebeslag te beperken kunt u een aantal dingen doen:

1 Als u bij voorbeeld een tweedimensionale tabel van 20 rijen en 20 kolommen wilt dimensioneren, bedenk dan dat DIM A(19,19) voldoende is (dus niet DIM A(20,20)).
2 Reserveer per veld niet meer bytes dan nodig is. Met DIM A(9,9) reserveert u 10 x 10 x 8 = 800 bytes geheugenruimte omdat A als dubbele-precisievariabele wordt beschouwd (8 bytes per variabele). Wilt u in de tabel uitsluitend enkele-precisievariabelen opslaan, gebruik dan DIM A!(9,9) (dat kost 400 bytes). Wilt u alleen integers opslaan, gebruik dan DIM A%(9,9); u reserveert dan maar 200 bytes.
3 De geheugenruimte voor de tabel blijft gereserveerd totdat u de computer uitschakelt of met de instructie ERASE de tabel verwijdert:

```
100 ERASE A
```

Dezelfde instructie kunt u gebruiken om een tabel opnieuw te dimensioneren, bij voorbeeld als u hem wilt uitbreiden:

```
10 DIM A(20)
20 ERASE A
30 DIM A(25)
```

Laat u in dit voorbeeld regel 20 weg, dan krijgt u de foutmelding "Redimensioned array in 30" omdat u probeerde tweemaal ruimte voor tabel A te reserveren. Regel 20 wist tabel A voordat de nieuwe tabel A wordt gedimensioneerd.

U kunt ook tabellen dimensioneren voor alfanumerieke variabelen (strings), bij voorbeeld: DIM A$(20).

2.7 Beslissingen

2.7.1 IF...THEN

Het verdere verloop van een programma kan afhankelijk worden gemaakt van het feit of al dan niet aan een bepaalde voorwaarde of conditie is voldaan. Hiervoor kan de instructie IF....THEN....ELSE worden gebruikt ("als....dan....anders"), zoals in het volgende voorbeeld:

```
10 INPUT A:INPUT B
20 IF A=B THEN PRINT "A=B" ELSE PR
INT "A<>B"
```

In regel 20 worden de waarden van A en B vergeleken. Met de instructie IF A = B onderzoekt MSX-BASIC of aan de voorwaarde is voldaan dat A en B gelijk zijn. Is aan die voorwaarde voldaan dan (THEN) print de computer "A = B" op het scherm; anders (ELSE) wordt "A < > B" op het scherm gezet (A is ongelijk aan B). Het laatste deel van regel 20, het woord ELSE en wat er op volgt, mag worden weggelaten. Bovendien kan regel 10 als volgt worden geschreven, dus met één INPUT-instructie:

```
10 INPUT A,B
20 IF A=B THEN PRINT "A=B"
```

Is nu aan de voorwaarde "A = B" voldaan, dan zet de computer "A = B" op het scherm en gaat hij verder naar de volgende programmaregel. Is niet aan de voorwaarde voldaan, dan gaat de computer direct door naar de volgende regel.
De instructie IF...THEN kan ook worden gebruikt om, afhankelijk van de conditie, naar een bepaalde programmaregel te springen:

```
10 INPUT A,B
20 IF A=0 THEN GOTO 50 ELSE IF A=B
THEN GOTO 40 ELSE GOTO 30
30 PRINT "A< >B:GOTO 20
40 PRINT "A=B":GOTO 20
50 END
```

U kunt het programma verlaten door voor A nul in te toetsen.
Regel 20 is een beetje dubbelop. Als u THEN gebruikt kunt u GOTO weglaten, en als u GOTO gebruikt kunt u THEN weglaten. Ook GOTO na ELSE is overbodig.
Regel 20 kan dus luiden: "IF A = 0 THEN 50 ELSE IF A = B THEN 40 ELSE 30" of "IF A = 0 GOTO 50 ELSE IF A = B GOTO 40 ELSE 30".

2.7.2 Relationele operatoren

Een "operator" is een symbool dat een bepaalde bewerking of berekening aangeeft. De rekenkundige operatoren +, -, *, / enz. hebben we al beschreven in § 2.4. Een ander type zijn de relationele operatoren, die worden gebruikt om twee waarden met elkaar te vergelijken. In de vorige paragraaf hebben we er al een paar gebruikt, in combinatie met de instructie IF....THEN ....(ELSE). Er zijn zes relationele operatoren:

>	groter dan
<	kleiner dan
=	gelijk aan
> < of < >	ongelijk aan
> = of = >	groter dan of gelijk aan
< = of = <	kleiner dan of gelijk aan

Als een IF...THEN-instructie zowel rekenkundige als relationele operatoren omvat, worden eerst de rekenkundige bewerkingen uitgevoerd, zoals in het volgende voorbeeld:

```
10 INPUT A,B
20 IF A MOD B=0 THEN PRINT"Rest =
0"
```

De instructie die op THEN volgt wordt alleen uitgevoerd

als het resultaat van de modulus-berekening A MOD B gelijk is aan nul.

2.7.3 Logische operatoren

Met de logische operator NOT kunnen we een voorwaarde als het ware omkeren, zoals in het volgende voorbeeld:

```
10 INPUT A,B
20 IF NOT A=B THEN PRINT "A<>B"
30 IF A<>B THEN PRINT "A<>B"
```

De regels 20 en 30 leiden tot hetzelfde resultaat.

Behalve NOT zijn er nog enkele logische operatoren, namelijk AND, OR, XOR, EQV en IMP. Bij alle bewerkingen met logische operatoren zijn er maar twee mogelijkheden: een bewering is WAAR of een bewering is ONWAAR. De bewering "A = 3 AND B = 5" is WAAR als A gelijk is aan 3 EN B gelijk is aan 5. De bewering is ONWAAR als één van de twee vergelijkingen ONWAAR is, dus als A ongelijk is aan 3, of als B ongelijk is aan 5. Voor de zes logische operatoren kunnen we de volgende "waarheidstabel" opstellen:

	Conditie 1	Conditie 2	Resultaat
AND	waar	waar	waar
	waar	onwaar	onwaar
	onwaar	waar	onwaar
	onwaar	onwaar	onwaar
OR	waar	waar	waar
	waar	onwaar	waar
	onwaar	waar	waar
	onwaar	onwaar	onwaar
XOR	waar	waar	onwaar
	waar	onwaar	waar
	onwaar	waar	waar
	onwaar	onwaar	onwaar
EQV	waar	waar	waar
	waar	onwaar	onwaar
	onwaar	waar	onwaar
	onwaar	onwaar	waar
IMP	waar	waar	waar
	waar	onwaar	onwaar
	onwaar	waar	waar
	onwaar	onwaar	waar
NOT	waar		onwaar
	waar		onwaar
	onwaar		waar
	onwaar		waar

Enkele voorbeelden:

```
10 IF 3>1 AND 3>2 THEN PRINT "WAAR"
20 IF 3>1 OR 6<7 THEN PRINT "WAAR"
30 IF 3>1 XOR 6<1 THEN PRINT "WAAR"
40 IF 4<3 EQV 6<7 THEN PRINT "WAAR"
50 IF 4<3 IMP 3>2 THEN PRINT "WAAR"
```

In alle voorbeelden is aan de voorwaarde voldaan en wordt de instructie na THEN uitgevoerd.

2.7.4 Het vergelijken van strings

Niet alleen numerieke, maar ook alfanumerieke constanten en variabelen (strings) kunnen worden gebruikt in een IF...THEN-instructie om te zien of aan een bepaalde conditie is voldaan. In dit geval worden de codes van de symbolen vergeleken (elk symbool heeft een bepaalde code; bij voorbeeld: A = 65, B = 66, a = 97 enz.). Een compleet overzicht van al deze codes vindt u in Appendix E.

Voorbeeld:

```
10 IF "A"<"B" THEN PRINT "A is klei
ner dan B"
20 IF A$<B$ THEN PRINT "De waarde v
an A$ is kleiner dan de waarde van
B$"
30 IF A$="John" THEN PRINT "De inho
ud van A$ is John"
40 IF "Johan"<"John" THEN PRINT "Co
rrect"
```

Numerieke en alfanumerieke constanten en variabelen kunnen echter niet met elkaar worden vergeleken.

2.7.5 Programmalussen

Wis het computergeheugen met het commando NEW en typ het volgende programma in:

```
NEW
10 CT=1
20 PRINT "MSX-BASIC"
30 CT=CT+1
40 IF CT<11 GOTO 20
RUN
```

U zult zien dat dit programma tienmaal het woord MSX-BASIC op het scherm zet. Er is echter een kortere manier om hetzelfde effect te krijgen:

```
NEW
10 FOR CT=1 TO 10
20 PRINT "MSX-BASIC"
30 NEXT CT
```

```
40 PRINT "Einde"
50 END
RUN
```

In regel 10 krijgt CT de waarde 1. In regel 20 wordt de tekst op het scherm gezet en in regel 30 wordt de waarde van CT met 1 opgehoogd. Als CT groter wordt dan 10, gaat het programma verder met regel 40. Zo lang CT dus kleiner dan of gelijk aan 10 is, zullen de regels 20 en 30 worden uitgevoerd.
In dit voorbeeld wordt CT elke keer opgehoogd met 1. Met de instructie STEP (= stap) kan CT met een andere waarde worden opgehoogd, elke keer dat de "programmalus" (de regels 20 en 30) wordt doorlopen. CT kan zelfs met een negatieve waarde worden "opgehoogd", en in plaats van een constante mag een variabele worden gebruikt, zoals in het volgende voorbeeld:

```
NEW
10 A=4
20 FOR CT=8 TO A STEP -2
30 PRINT "MSX-BASIC"
40 NEXT CT
RUN
```

In dit voorbeeld wordt de waarde van CT in feite verlaagd met 2, elke keer dat de programmalus wordt doorlopen. Dit betekent dat de beginwaarde van CT hoger moet zijn dan de eindwaarde. U zult zien dat het bovenstaande programma driemaal MSX-BASIC op het scherm zet.

2.7.6 Sorteren

Het volgende programma zet de getallen, die door de gebruiker worden ingetoetst, in de juiste volgorde. De gebruiker moet aan de computer opgeven hoeveel getallen gesorteerd moeten worden, en vervolgens de betrokken getallen intoetsen. De getallen worden opgeslagen in een tabel en op volgorde gezet door middel van de zogenaamde binaire sorteermethode.
De instructie REM (van Remark = opmerking) in de regels 75 en 245 is bedoeld om commentaar in een programma op te nemen. Alles dat achter REM staat wordt door MSX-BASIC niet uitgevoerd; het programma gaat verder met de volgende programmaregel.

```
NEW
10 DIM G(100)
20 INPUT "Hoeveel getallen (tussen
2 en 100)";A
30 IF (A<2)OR(A>100) GOTO 20
40 FOR I=1 TO A
50 PRINT "Nummer";I;
60 INPUT G(I)
70 NEXT I
75 REM Als I kleiner is dan de inh
oud van A, gaat het programma verd
er met regel 50
80 PRINT "Ik ben nu aan het sorter
en"
90 FOR I=2 TO A
100 X1=1:X2=I
110 X3=X1+INT((X2-X1)/2)
120 IF G(I)<G(X3) THEN 150
130 IF G(I)>G(X3) THEN 170
140 X2=X3
150 IF X2=X3 THEN 190
160 X2=X3:GOTO 110
170 IF X1=X3 THEN 190
180 X1=X3:GOTO110
190 H=G(I)
200 FOR J=I TO X2+1 STEP-1
210 G(J)=G(J-1)
220 NEXT J
230 G(X2)=H
240 NEXT I
245 REM Als I kleiner is dan de in
houd van A, gaat het programma ver
der met regel 100
250 PRINT "Dit zijn de getallen in
volgorde"
260 FOR I=1 TO A
270 PRINT G(I);
280 NEXT I
RUN
```

Als u dit programma laat uitvoeren, zult u zien dat de getallen netjes in de juiste volgorde worden gezet. Werkt het programma niet naar behoren, controleer het dan zorgvuldig.

2.7.7 Programma's met een menu

Programma's die verschillende functies moeten kunnen uitvoeren, beginnen vaak met een keuzemenu. Het menu geeft een overzicht van de keuzemogelijkheden en het desbetreffende programmadeel kan worden aangeroepen door het nummer in te toetsen dat bij die keuze hoort. Hieronder een voorbeeld van een menuprogramma.

```
NEW
10 CLS:PRINT "Keuzemenu"
20 PRINT "1 = Programmaregel 100"
30 PRINT "2 = Programmaregel 200"
40 PRINT "3 = Einde programma"
50 INPUT "Uw keus";A
60 ON A GOTO 100,200,300:GOTO 50
100 PRINT "Dit is programmaregel
100":GOTO 10
200 PRINT "Dit is programmaregel
200":GOTO 10
300 END
```

De meest interessante regel in dit programma is regel 60. Als de waarde van A gelijk is aan 1, gaat het programma verder met regel 100. Is de waarde 2, dan gaat het programma verder met regel 200 en bij waarde 3 gaat het verder met regel 300. Is de waarde groter dan 3, dan wordt regel 50 opnieuw uitgevoerd.

2.7.8 Het afhandelen van fouten

MSX-BASIC maakt het mogelijk fouten in het programma op te sporen en af te handelen. Hier is een voorbeeld:

```
NEW
10 ON ERROR GOTO 50
20 INPUT "Welk getal";A%
30 END
50 IF ERR=6 THEN PRINT "Het getal
moet tussen -32768 en +32767 ligge
n":RESUME 0
```

De instructie in regel 10 heeft tot gevolg dat het programma naar regel 50 springt als het een fout ontdekt. Toetst de gebruiker, als antwoord op de vraag in regel 20, een getal in dat kleiner is dan -32768 of groter is dan +32767, dan geeft MSX-BASIC een "overflow"-fout omdat A% een integer-variabele is. Deze fout heeft nummer 6 (zie Appendix A). Regel 50 onderzoekt of fout nummer 6 inderdaad de oorzaak is van de foutmelding. Is dat het geval, dan wordt de tekst van regel 50 op het scherm gezet. De RESUME-instructie wordt gebruikt om het programma opnieuw te starten, te beginnen met regel 0. In regel 60 zal MSX-BASIC de uitvoering van het programma afbreken als een fout met een ander nummer dan foutcode 6 optreedt.

Met MSX-BASIC is het ook mogelijk uw eigen foutcodes te definiëren. Het enige waar u aan moet denken is dat het nummer van de foutcode hoger moet zijn dan het hoogste nummer dat MSX-BASIC zelf gebruikt (zie Appendix A). Een voorbeeld:

```
NEW
10 ON ERROR GOTO 100
20 INPUT "Programma beëindigen";A$
30 IF A$="ja" THEN ERROR 250
40 PRINT "Programma wordt voortgez
et":GOTO 20
50 END
100 IF ERR=250 THEN INPUT "Weet u
het zeker";A$:IF A$="ja" THEN RES
UME 50 ELSE RESUME 20
110 ON ERROR GOTO 0
```

In programmaregel 30 wordt foutcode 250 gebruikt als de variabele A$ gelijk is aan "ja". In dat geval wordt regel 100 uitgevoerd, zoals in regel 10 is aangegeven. Regel 100 onderzoekt of de fout wordt veroorzaakt door foutcode 250. Is dat het geval, dan zal MSX-BASIC u om bevestiging vragen. Antwoordt u bevestigend, dan gaat het programma door met regel 50 als gevolg van de instructie RESUME 50. Antwoordt u ontkennend, dan maakt de instructie RESUME 20 dat het programma verder gaat met regel 20.

2.8 Subroutines

2.8.1 Inleiding

Laten we aannemen dat het volgende programma is opgeslagen in het geheugen van de computer:

```
10 INPUT A:INPUT B
20 C=A*100/B:PRINT A;
30 PRINT "is"C"procent van";
40 PRINT B
50 C=B*100/A:PRINT B;
60 PRINT "is"C"procent van";
70 PRINT A
80 END
```

Het zal u zonder twijfel zijn opgevallen dat programmaregel 60 identiek is aan regel 30. Daarom kunt u ook een speciale programmaregel gebruiken en in de regels 30 en 60 met GOSUB naar die regel verwijzen:

```
10 INPUT A:INPUT B
20 C=A*100/B:PRINT A;
30 GOSUB 90
40 PRINT B
50 C=B*100/A:PRINT B;
60 GOSUB 90
70 PRINT A
80 END
90 PRINT "is"C"procent van";
100 RETURN
```

Als MSX-BASIC regel 30 bereikt, zal het programma doorgaan naar regel 90. De RETURN-instructie laat het programma terugkeren naar regel 40. Hetzelfde gebeurt als het programma regel 60 bereikt, zij het dat nu van regel 100 wordt teruggesprongen naar regel 70.
De instructies in de programmaregels 90 en 100 noemt men een "subroutine". Wanneer MSX-BASIC als gevolg van de GOSUB-instructie op regel 90 komt, "herinnert" hij zich waar hij vandaan kwam. Daarom zal hij in dit voorbeeld na de RETURN-instructie terugkeren naar regel 40 of regel 70, de regel die onmiddellijk volgt op de regel waar hij vandaan kwam.
Regel 80 van ons voorbeeldprogramma is van bijzonder

belang. Vergeet u de END-instructie, zodat regel 90 direct op regel 70 volgt, dan zal MSX-BASIC de foutmelding "RETURN without GOSUB" geven als hij op regel 100 komt.

2.8.2 Subroutine op verzoek

Een subroutine kan ook worden aangeroepen door middel van:

1 de inhoud van een variabele
2 de tijd
3 een of meer functietoetsen
4 de STOP- + de CTRL-toets
5 een botsing tussen twee sprites
6 de actietoets van een spelregelaar

De laatste twee mogelijkheden worden uitvoeriger beschreven in de paragrafen 2.12 en 2.14.
Een subroutine aanroepen naar aanleiding van de inhoud van een variabele gebeurt met het woord ON.
Een voorbeeld:

```
10 INPUT A
20 IF A<0 OR A>255 GOTO 10
30 ON A GOSUB 100,200,300
40 END
100 PRINT "Subroutine 100":RETURN
200 PRINT "Subroutine 200":RETURN
300 PRINT "Subroutine 300":RETURN
```

In dit voorbeeld wordt regel 100 uitgevoerd als A gelijk is aan 1. De subroutine in regel 200 wordt uitgevoerd als A = 2 en die in regel 300 als A = 3. De waarde van de variabele A in regel 30 kan niet negatief zijn of groter zijn dan 255 omdat dit in regel 20 wordt onderzocht. Is dat wel het geval, dan zal MSX-BASIC terugkeren naar regel 10 en kan een nieuwe waarde worden ingevoerd.
In plaats van een variabele A kan ook een wiskundige formule worden opgegeven, bij voorbeeld:

```
30 ON A+1 GOSUB 100,200,300
```

Is variabele A = 1, dan is A + 1 = 2 en zal de subroutine op regel 200 worden uitgevoerd. Voor dergelijke berekeningen geldt dezelfde regel, dat de uitkomst niet

negatief of groter dan 255 mag zijn. Als de variabele A of de uitkomst van de berekening groter is dan het aantal GOSUB-regels, dan gaat MSX-BASIC door met de volgende instructie. Is bij voorbeeld A = 3, dan zal regel 40 worden uitgevoerd omdat A + 1 = 4 en omdat slechts drie subroutineregels zijn ingevoerd achter GOSUB in regel 30.

Het aanroepen van een subroutine, afhankelijk van de tijd, het indrukken van een functietoets of het indrukken van STOP + CTRL, kan met behulp van de volgende programmaregels:

```
10 ON STOP GOSUB 100
20 ON INTERVAL=250 GOSUB 200
30 ON KEY GOSUB 300
40 STOP ON:INTERVAL ON:KEY(1) ON
50 GOTO 50
100 PRINT "Einde":END
200 PRINT "Alweer 5 seconden voorb
ij"
210 RETURN
300 PRINT "Functietoets 1 ingedruk
t"
310 RETURN
```

In dit programmavoorbeeld is in regel 10 aangegeven naar welke subroutine moet worden gesprongen als de STOP- en de CTRL-toets tegelijk worden ingedrukt. In regel 20 staat de tijd die moet verstrijken voordat naar de subroutine op regel 200 wordt gesprongen (in dit voorbeeld 250*1/50 s = 5 seconden). En in regel 30 staat de subroutine waar naartoe moet worden gesprongen als functietoets 1 wordt ingedrukt.
In regel 40 worden deze drie functies geactiveerd. Regel 50 maakt de indruk volkomen overbodig te zijn, omdat het lijkt alsof MSX-BASIC tot in de eeuwigheid deze regel zal blijven uitvoeren. Terwijl hij hiermee bezig is, zal hij zich echter voortdurend afvragen of één van de geactiveerde toetsen is ingedrukt of dat het tijdinterval van 5 seconden is verstreken. In dat geval zal hij de overeenkomstige subroutine uitvoeren.

Merk op dat de subroutine in regel 100 niet is afgesloten met RETURN, maar met END. Anders zou u het programma alleen nog maar kunnen onderbreken door de computer uit te schakelen.

2.8.3 Welke dag is het ?

In het volgende voorbeeld berekent MSX-BASIC welke dag van de week het is als u een bepaalde datum intoetst.

```
NEW
10 DATA zondag,maandag,dinsdag,woe
nsdag,donderdag,vrijdag,zaterdag
20 PRINT "Voer een datum in"
30 INPUT "Dag (1...31)";DD
40 IF DD<1 OR DD>31 GOTO 30
50 INPUT "Maand (1...12)";MM
60 IF MM<1 OR MM>12 GOTO 50
70 INPUT "Jaar (1...3000)";YY
80 IF YY<1 OR YY>3000 GOTO 70
90 IF MM>2 GOTO 110
100 YY=YY-1:MM=MM+12
110 H1=INT(YY/100):H2=YY-H1*100
120 H3=INT(2.6001*(MM-2)-0.2)+DD+H
2+INT(H2/4)+INT(H1/4)-2*H1
130 H3=H3-INT(H3/7)*7+1
135 REM Regelnummers 100 tot 130 z
ijn de wiskundige berekeningen
140 RESTORE
150 FOR I=1 to H3:READ H$:NEXT I
160 PRINT "Deze datum valt op een
";H$
170 END
RUN
```

2.9 Functies

2.9.1 Inleiding

Behalve commando's en instructies kent MSX-BASIC een groot aantal functies. Deze leveren de uitkomst van een berekening met een constante of een variabele. Functies kunnen alleen worden gebruikt in combinatie met een instructie. Twee voorbeelden:

```
NEW
10 PRINT SIN(5)
RUN
-.9589242746631

NEW
10 X=5
20 PRINT SIN(X)
RUN
-.9589242746631
```

Een volledig overzicht van de beschikbare functies vindt u in het Naslagwerk voor Programmeurs. Ze kunnen worden verdeeld in de volgende groepen:

Type functie:

Wiskundig	**String**	**Conversie**	**Input**	**Output**	**Diversen**
ABS	ASC	CDBL	INKEY$	CHR$	CSRLIN
ATN	LEFT$	CINT	INPUT$	POS	ERL
COS	LEN	CSNG		LPOS	ERR
EXP	MID$	FIX		SPC	FRE
INT	RIGHT$			TAB	PEEK
LOG	STR$				RND
SGN	STRING$				USR
SIN	VAL				VARPTR
SQR	BIN$				POINT
TAN	HEX$				VPEEK
	OCT$				STICK
	INSTR				STRIG
					PDL
					PAD
					PLAY
					EOF

U kunt ook uw eigen functies definiëren met DEF FN (zie § 2.4.6).

2.9.2 Een willekeurig getal

De functie RND wordt dikwijls gebruikt om MSX-BASIC een willekeurig getal te laten bepalen. Dat getal ligt tussen 0 en 1. Wilt u de computer een willekeurig getal tussen 1 en 6 laten opwekken (bij voorbeeld om het werpen met een dobbelsteen na te bootsen), dan moet het willekeurige getal met 6 worden vermenigvuldigd, waarna er 1 bij wordt opgeteld. Als voorbeeld:

```
NEW
10 A%=INT(RND(1)*6+1)
20 PRINT A%
RUN
```

In dit voorbeeld zijn in regel 10 twee functies in combinatie met een enkele instructie gebruikt. De functie INT maakt van het willekeurige getal een geheel getal (integer).

2.9.3 Lottogetallen

Het volgende programma kiest zes getallen voor het invullen van het lottoformulier:

```
NEW
10 X=0
20 FOR I=1 TO 6
30 A%=INT(RND(1)*41+1)
40 FOR J=1 TO 5
50 IF A%=X(I) GOTO 30
60 NEXT J
70 X(I)=A%
80 NEXT I
90 PRINT "De lottogetallen zijn:";
100 FOR I=1 to 6:PRINT X(I);:NEXT
I:PRINT
110 INPUT "Nog eens (J/N)";A$
120 IF A$="J" OR A$="j" GOTO 10
130 END
```

Laat dit programma lopen en kijk welke lottogetallen MSX-BASIC voor u bedenkt. Als het programma niet goed werkt, controleer dan zorgvuldig of u alle instructies goed hebt ingetoetst.

Opmerking: elke keer als u het programma stopt met CTRL + STOP en het daarna met RUN opnieuw laat lopen, genereert MSX-BASIC dezelfde reeksen van 6 getallen, waarvan de eerste uit de getallen 25, 5, 32, 24, 31 en 8 bestaat. Het verdient dus aanbeveling de vraag in regel 110 een aantal keren met "ja" te beantwoorden, anders krijgt u niet alleen elke week dezelfde reeks getallen op uw lottoformulier, maar andere gebruikers van dit MSX-programma ook.

2.9.4 Vrije geheugenruimte

Met de functie FRE kunt u opvragen hoeveel vrije geheugenruimte (RAM) nog beschikbaar is. Dat gaat als volgt (in de directe stand):

```
PRINT FRE(0)
 nnnnn
```

Dit betekent dat u nog nnnnn bytes vrije geheugenruimte tot uw beschikking hebt.
Als u wilt weten hoeveel geheugenruimte er nog is voor alfanumerieke variabelen (strings), kunt u de volgende instructie gebruiken:

```
PRINT FRE("")
 xxx
```

Dit betekent dat u nog xxx bytes beschikbaar hebt voor het opbergen van stringvariabelen.

2.10 MSX-BASIC en het scherm

2.10.1 Inleiding

Met MSX-BASIC kunt u het beeldscherm op vier verschillende manieren gebruiken:

1 Tekst-stand 1
2 Tekst-stand 2
3 Grafische stand 1
4 Grafische stand 2

2.10.2 Tekststand 1

In deze stand biedt het scherm plaats aan 24 regels van elk 40 posities. Elke positie wordt bepaald door een X-coördinaat (die de kolom aangeeft) en een Y-coör-

dinaat (die de regel aangeeft). De kleinste X-coördinaat is 0; deze komt overeen met de meest linkse kolom op het scherm. De grootste X-coördinaat is 39, de meest rechtse kolom.
De Y-coördinaten liggen tussen 0 (bovenste regel) en 23 (onderste regel).
Met de instructie WIDTH n kunt u de computer vertellen hoeveel posities u op een regel wenst, waarbij n tussen 1 en 40 moet liggen.
Met de instructie LOCATE kunt u de cursor naar elke gewenste positie op het scherm brengen.
Tekststand 1 kunt u kiezen met de instructie SCREEN 0.
Deze drie instructies zijn in het volgende programma verwerkt:

```
NEW
10 SCREEN 0
20 WIDTH 40
30 LOCATE 15,10:PRINT "Hallo"
RUN
```

In het bovenstaande voorbeeld wordt in regel 10 tekststand 1 geactiveerd. Regel 20 opent de mogelijkheid 40 tekens op een regel te plaatsen. In regel 30 wordt de cursor naar de 15e positie (zestiende kolom) van regel 10 (elfde regel) gebracht, waar het woord "Hallo" wordt afgedrukt.
Als MSX-BASIC in de directe stand staat of als hij bezig is met het afhandelen van een INPUT-instructie, staat het scherm altijd in tekststand 1 of 2.

2.10.3 Tekststand 2

In deze tekststand kunnen ten hoogste 32 tekens op een regel worden gezet. Elke positie op het scherm wordt weer bepaald door een X-coördinaat voor de kolom en een Y-coördinaat voor de regel. De X-coördinaat ligt tussen 0 en 31, de Y-coördinaat tussen 0 en 23. Het maximumaantal tekens per regel kan weer worden bepaald met de instructie WIDTH n, waarbij n dus tussen 0 en 31 moet liggen. Met LOCATE kan de cursor naar een willekeurige plaats op het scherm worden gebracht.
Deze tekststand wordt geactiveerd met de instructie SCREEN 1. Hiervan een voorbeeld:

```
NEW
10 SCREEN 1
20 WIDTH 32
30 LOCATE 15,10:PRINT "Hallo"
RUN
```

Dit programma leidt tot precies hetzelfde resultaat als het voorbeeld uit § 2.10.2. In regel 10 hebben we nu echter tekststand 2 gekozen en in regel 20 hebben we de mogelijkheid geopend 32 tekens per regel weer te geven. Als u de tekens op het scherm in tekststand 1 niet goed kunt lezen, gebruik dan tekststand 2.
Merk op dat er in tekststand 2 een cyaankleurige rand om het beeld aanwezig is. U kunt de kleur van deze rand, van de achtergrond in het midden en van de tekens veranderen met de instructie COLOR (zie § 2.11).

Ga nu terug naar tekststand 1 door SCREEN 0:WIDTH 40 in te toetsen.

2.10.4 Grafische stand 1

In deze stand is het scherm horizontaal in 256 puntjes en verticaal in 192 puntjes verdeeld. De positie van elk puntje wordt weer bepaald door een X-coördinaat (de kolom) en een Y-coördinaat (de regel). De X-coördinaat kan waarden van 0 tot 255 hebben en de Y-coördinaat waarden van 0 tot 191.
In deze stand kunnen alleen grafische instructies worden gebruikt om een beeld op het scherm te zetten.
Grafische stand 1 wordt geactiveerd met de instructie SCREEN 2. De grafische instructies worden verklaard in § 2.11.
Eerst een voorbeeld:

```
NEW
10 SCREEN 2
20 CIRCLE (185,115),10
30 GOTO 30
RUN
```

Dit programma maakt dat uw computer een cirkelvormige figuur op het scherm tekent, met de punten 185

(X-coördinaat) en 115 (Y-coördinaat) als middelpunt. Regel 30 is hierbij van bijzonder belang. Deze regel wordt voortdurend doorlopen. Laat u regel 30 weg, dan keert MSX-BASIC na het uitvoeren van regel 20 terug naar de directe stand, waarbij hij automatisch terugschakelt naar tekststand 1. MSX-BASIC werkt echter zo snel, dat u het tekenen van de figuur niet zou zien. Regel 30 maakt dat de computer wacht totdat u het programma onderbreekt door CTRL en STOP tegelijk in te drukken.

2.10.5 Grafische stand 2

Als grafische stand 2 is geactiveerd met de instructie SCREEN 3, is het scherm op precies dezelfde manier ingedeeld als in grafische stand 1. Het verschil is dat in grafische stand 2 alle beelden zijn opgebouwd uit blokjes van 4 x 4 beeldpunten.
In deze stand kunnen alleen grafische instructies worden gebruikt. Meer details hierover vindt u in § 2.11.
Eerst nog een voorbeeld:

```
NEW
10 SCREEN 3
20 LINE (125,10)-(200,85)
30 GOTO 30
RUN
```

In deze oefening zal een lijn worden getrokken van punt 125 (X-coördinaat), 10 (Y-coördinaat) naar punt 200 (X), 85 (Y). De lijn is opgebouwd uit kleine blokjes.
U kunt het programma onderbreken door CTRL en STOP tegelijk in te drukken.
Laten we nu eens kijken wat er gebeurt als we regel 10 veranderen in:

```
10 SCREEN 2
```

Als u het verschil gezien hebt, onderbreek dan weer het programma door de CTRL- en de STOP-toets tegelijk in te drukken.

2.10.6 Wissen van het scherm

U kunt het scherm schoonmaken met de instructie CLS. Dat kan zowel in de directe stand als in een programma.

U kunt in de directe stand natuurlijk ook de CLR-toets gebruiken (met SHIFT).

2.11 Instructies voor kleur en grafische beelden

2.11.1 Inleiding

Met MSX-BASIC kunt u gebruik maken van 16 kleuren, genummerd van 0 tot en met 15. De kleur van de rand, de voorgrond en de achtergrond kan in tekststand 2 en grafische stand 1 en 2 afzonderlijk worden geprogrammeerd. U vindt een volledige lijst van de kleuren en de bijbehorende codenummers in Appendix C.

De kleur met codenummer 0 (transparant) is van bijzonder belang. Als u deze kleurcode gebruikt, zullen de niet-transparante kleuren alleen zichtbaar zijn op de plaatsen waar transparant is geprogrammeerd. Dit betekent dat transparant alleen als voorgrondkleur wordt gebruikt. Als zowel de voorgrond als de achtergrond transparant is, zal het desbetreffende deel van het scherm zwart blijven.

De kleuren kunnen worden gekozen met de instructie COLOR. Wis het computergeheugem met het commando NEW en toets het volgende voorbeeld in:

```
10 SCREEN 1
20 COLOR 1,3,6
30 PRINT "Hallo"
RUN
```

In dit voorbeeld wordt de voorgrond, dat wil zeggen de letters, zwart (kleur 1), de achtergrond lichtgroen (kleur 3) en de rand donkerrood (kleur 6).
Als u dit hebt gezien, toets dan in:

```
SCREEN 0:COLOR 15,4,7
```

COLOR 15,4,7 kunt u door middel van functietoets 6 met één toetsindruk intikken.
Als u uw MSX-computer inschakelt, zijn de standaardkleuren in tekststand 1: wit als voorgrondkleur (kleur 15) en donkerblauw als achtergrondkleur (kleur 4). Bij de andere drie typen schermbeelden is de rand lichtblauw (cyaan, kleur 7).

2.11.2 Het trekken van een lijn

Met de instructie LINE kunt u op het scherm een lijn tussen twee punten trekken. Deze punten moeten worden opgegeven met hun X- en Y-coördinaat. Dat kan zowel een absolute als een relatieve waarde zijn. Absoluut wil zeggen dat u de X- en de Y-coördinaat opgeeft ten opzichte van het punt linksboven op het scherm, waarvoor zowel X als Y nul is. Een punt in kolom 11 en regel 16 heeft bij voorbeeld de absolute coördinaten (10,15). Relatief betekent dat de X- en de Y-coördinaat worden opgegeven ten opzichte van een ander punt op het scherm. Voor het aangeven van de relatieve coördinaten dient u het woord STEP te gebruiken. Nemen we aan dat de laatste absolute coördinaten die u hebt gebruikt (3,6) waren, dan komt het punt in kolom 11 regel 16 (absoluut: (10,15)) overeen met STEP (7,9).

```
NEW
10 SCREEN 2
20 LINE (125,10)-STEP(75,75),1
30 GOTO 30
RUN
```

In dit voorbeeld zijn de coördinaten van het tweede punt opgegeven ten opzichte van het eerste punt. Het cijfer 1 aan het einde van regel 20 bepaalt de kleur van de lijn (in dit geval kleur 1 = zwart).
U had regel 20 ook als volgt kunnen schrijven:

```
20 LINE (125,10)-(200,85),1
```

Voegt u na het kleurnummer (1) een komma en de letter B toe, dan zal MSX-BASIC een rechthoek op het scherm tekenen, met de opgegeven coördinaten als twee van de hoekpunten. De oorspronkelijke lijn is een diagonaal van deze rechthoek, maar deze is niet meer zichtbaar. Onderbreek het programma met de toetsen CTRL en STOP, verander regel 20 als volgt en laat het programma lopen:

```
20 LINE (125,10)-STEP(75,75),1,B
```

Toetst u in regel 20 ",BF" in na het kleurnummer, dan zal MSX-BASIC eveneens een rechthoek op het scherm tekenen, maar deze opvullen met de aangegeven kleur

(in dit geval 1 = zwart). Verander nu regel 20 als volgt en laat het programma weer lopen:

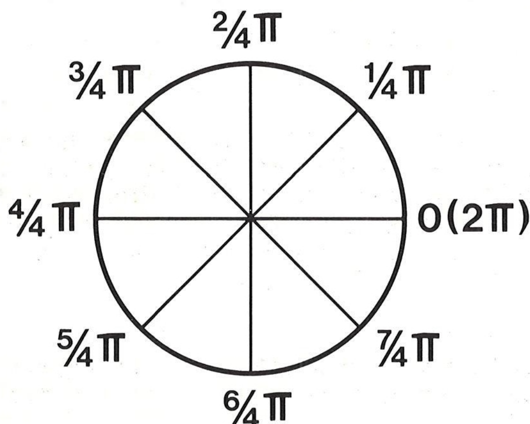
```
20 LINE (125,10)-STEP(75,75),1,BF
```

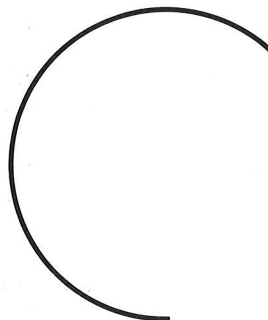
2.11.3 Het tekenen van een cirkel

Zoals u inmiddels weet is het mogelijk een cirkel op het scherm te tekenen met de instructie CIRCLE als u de X- en Y-coördinaat van het middelpunt opgeeft. Dit mogen, net als in § 2.11.2, zowel absolute als relatieve coördinaten zijn.
Behalve de coördinaten van het middelpunt moeten ook de straal (in aantal beeldpunten) en het kleurnummer worden opgegeven, zoals in het volgende voorbeeld:

```
NEW
10 SCREEN 2
20 CIRCLE (90,80),20,1
30 GOTO 30
RUN
```

In dit voorbeeld wordt een cirkel getekend met de kleur zwart (kleurnummer 1), met een straal van 20 beeldpunten (de diameter is dus 40 beeldpunten) en met het middelpunt op X = 90 en Y = 80.
U kunt ook een deel van een cirkel tekenen door begin- en eindpunt op te geven. Daarbij wordt de cirkelomtrek verdeeld in radialen, zoals in de volgende afbeelding is aangegeven:





Zoals bekend is de omtrek circa 3,1416 maal de straal van een cirkel. Wiskundig geformuleerd: f (pi) = 3,1416. Laten we aannemen dat u een deel van een cirkel wilt tekenen, zoals in de rechterafbeelding hierboven. Dan dient u regel 20 in het laatste programma als volgt te veranderen:

```
20 CIRCLE
(90,80),20,1,0.7854,4.7124
```

Het beginpunt van de cirkelboog wordt als volgt berekend: 3.1416 x ¼ = 0.7854. Het eindpunt kunt u berekenen met: 3.1416 x 6/4 = 4.7124.

Begin- en eindpunt van de cirkelboog kunt u met het middelpunt verbinden door voor de gevonden waarden een minteken te plaatsen. Onderbreek het programma met CTRL + STOP en verander regel 20 als volgt:

```
20 CIRCLE (90,80),20,1,-0.7854,-
4.7124
```

U kunt natuurlijk ook een van de eindpunten met het middelpunt verbinden.

2.11.4 Het tekenen van een ellips

MSX-BASIC kan ook een ellips op het scherm tekenen. Om dit te bereiken dient u een getal in te voeren dat de verhouding aangeeft tussen horizontale en verticale straal van de ellips. Onderbreek het programma weer en verander regel 20 als volgt:

```
20 CIRCLE (90,80),20,1,,,2
```

Zorg ervoor dat u het juiste aantal komma's hebt ingetoetst voordat u het programma laat lopen, anders zal MSX-BASIC het cijfer 2, dat de verhouding tussen de beide stralen aangeeft, beschouwen als het beginpunt van een cirkel. In dit voorbeeld is de verticale straal 20 en de horizontale straal 20 : 2 = 10.

2.11.5 Opvullen met kleur

MSX-BASIC maakt het mogelijk een figuur, afgebakend door lijnen, op te vullen met een kleur. Voor dit doel

gebruikt u de instructie PAINT. Tegelijk met deze instructie moeten de kleur (het kleurnummer) en de positie van het in te kleuren vlak worden opgegeven in de vorm van X- en Y-coördinaten. De coördinaten mogen weer absoluut en relatief zijn.
Vanuit het punt dat door de opgegeven coördinaten wordt bepaald, gaat MSX-BASIC op zoek naar een vlak dat aan alle kanten wordt begrensd door lijnen met dezelfde kleur als in de PAINT-instructie is opgegeven. Dat vlak zal dan in de aangegeven kleur worden opgevuld. Probeer maar eens het volgende voorbeeld:

```
NEW
10 SCREEN 2
20 CIRCLE (50,80),30,1
30 CIRCLE (90,80),30,1
40 PAINT (70,80),1
50 GOTO 50
RUN
```

In dit voorbeeld tekent MSX-BASIC twee zwarte cirkels, die elkaar gedeeltelijk overlappen. Met de instructie PAINT wordt de cursor in het gebied geplaatst waar de cirkels elkaar overlappen en dat gebied wordt opgevuld met kleur 1.
Onderbreek het programma en verander regel 20 als volgt:

```
20 CIRCLE (50,80),30,6
```

Laat nu het programma weer lopen. De PAINT-instructie zal nu de cirkel van regel 30 geheel opvullen met zwart omdat dit het enige vlak is dat is omsloten door zwarte lijnen.
Merk op dat de snijpunten van twee lijnen altijd de kleur hebben van de laatst getekende lijn. In ons voorbeeld vormt de rode cirkel van regel 20 niet langer een ononderbroken lijn. Probeert u deze cirkel op te vullen, dan heeft dat grote consequenties. Onderbreek het programma en verander regel 40 als volgt:

```
40 PAINT (70,80),6
```

Toets hierna RUN in. MSX-BASIC zal nu het hele mid-

denveld rood kleuren omdat er geen vlak is dat geheel is omsloten door rode lijnen. De snijpunten van beide cirkels vormen "lekken". Stop het programma weer met de toetsen CTRL en STOP.
In de grafische stand 2 kan MSX-BASIC ook zoeken naar een gesloten lijn in een bepaalde kleur, zoals in het volgende voorbeeld:

```
NEW
10 SCREEN 3
20 LINE (125,10)-STEP(75,75),15,B
30 PAINT (150,15),6,15
40 GOTO 40
RUN
```

U ziet dat de rechthoek, gevormd door de witte lijnen uit regel 20, rood wordt gekleurd door de instructie in regel 30. Hieruit blijkt dat in grafische stand 2 de kleur, waarmee een vlak wordt opgevuld, niet dezelfde hoeft te zijn als de kleur van de lijnen die het vlak insluiten.
Stop het programma met de toetsen CTRL en STOP.

2.11.6 Het plaatsen van een punt

De instructie PSET kunt u gebruiken om een enkel beeldpunt in een bepaalde kleur op het scherm te zetten. Met de instructie PRESET kunt u het weer laten verdwijnen, zoals in het volgende voorbeeld:

```
NEW
10 SCREEN 2
20 CIRCLE (80,80),20,1
30 FOR I=1 TO 1000
40 PSET (80,80),15
50 PRESET (80,80)
60 NEXT
RUN
```

In dit voorbeeld wordt het middelpunt van de cirkel (regel 20) aangegeven door een knipperende beeldpunt. In grafische stand 2 (SCREEN 3) bestaat zo'n punt uit 4 x 4 beeldpunten.

2.11.7 Tekenen op het scherm

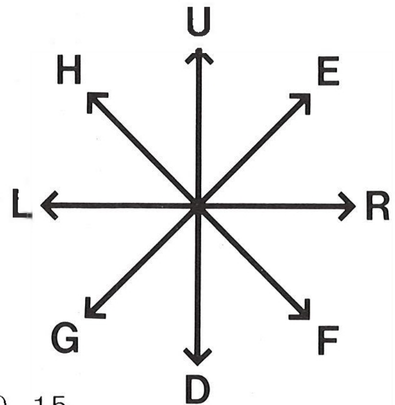
Er zijn nog meer manieren om tekeningen op het

scherm te maken. Ook met de instructie DRAW kunt u MSX-BASIC op het scherm laten tekenen. DRAW kent verscheidene sub-instructies, die we afzonderlijk zullen beschrijven.

2.11.8 Het tekenen van een lijn

Uitgaande van de positie van de cursor, kunt u een lijn trekken in acht verschillende windrichtingen. De letter n geeft de lengte van de lijn in beeldpunten aan:

Un = noord (Up)
Rn = oost (Right)
Dn = zuid (Down)
Ln = west (Left)
En = noordoost
Fn = zuidoost
Gn = zuidwest
Hn = noordwest



Een voorbeeld:

```
NEW
10 SCREEN 2
20 PSET (80,80),15
30 DRAW "E50"
40 GOTO 40
RUN
```

U zult zien dat MSX-BASIC een schuine lijn op het scherm tekent, in noordoostelijke richting. Merk op dat alles dat achter DRAW staat wordt beschouwd als een string en dus tussen aanhalingstekens moet staan.

Om een lijn vanuit de cursor naar een ander punt op het scherm te tekenen moet na DRAW de toevoeging "MX,Y" worden opgenomen, waarin X en Y de coördinaten zijn van het eindpunt van de lijn. De coördinaten mogen weer absoluut of relatief zijn. Als u voor de coördinaten relatieve waarden wilt opgeven, voeg dan aan X en Y een plus- of een minteken toe. De volgende twee voorbeelden hebben hetzelfde effect:

```
NEW
10 SCREEN 2
```

```
20 PSET (80,80),15
30 DRAW "M90,75"
40 GOTO 40
RUN
```

Stop nu het programma, verander regel 30 als volgt en laat het programma lopen:

```
30 DRAW "M+10,-5"
```

2.11.9 De subcommando's "N" en "B"

De subcommando's uit de vorige paragraaf kunt u laten voorafgaan door de subcommando's "N" en "B". Subcommando "N" maakt dat de cursor na het trekken van een lijn, dus na het uitvoeren van de DRAW-instructie, terugkeert naar het beginpunt van die lijn. Onderbreek het programma en typ het volgende voorbeeld in:

```
NEW
10 SCREEN 2
20 PSET (80,80),15
30 DRAW "R10D10"
40 GOTO 40
RUN
```

Met dit programma zal MSX-BASIC vanuit de positie X = 80, Y = 80 eerst een lijn van 10 beeldpunten naar rechts (R10) en vandaar een lijn naar beneden (D10) tekenen.
Onderbreek nu het programma en verander regel 30 als volgt:

```
30 DRAW "NR10D10"
```

Toets nu weer RUN in en u zult merken dat de computer nu eerst de lijn naar rechts tekent (R10), waarna de cursor terugkeert naar het beginpunt van die lijn, om daarna de lijn naar beneden (D10) te tekenen. Wilt u dat de cursor na het tekenen van lijn D10 opnieuw terug gaat naar het beginpunt, dan wordt regel 30:

```
30 DRAW "NR10ND10"
```

Het subcommando "N" moet dus voor elke lijn worden herhaald als u wilt dat de cursor terug gaat naar het beginpunt.

Het subcommando "B" heeft tot gevolg dat de cursor naar het gewenste punt gaat, zonder dat een lijn wordt getekend. Dit kunt u proberen met het volgende programma:

```
NEW
10 SCREEN 2
20 DRAW "BM80,80R10D10"
30 GOTO 30
RUN
```

Voor de betekenis van de letter "M": zie § 2.11.8.

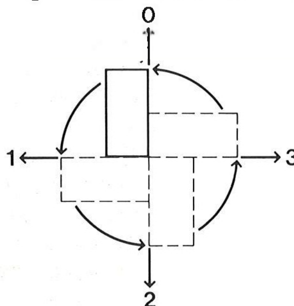
2.11.10 Subcommando voor kleur

Met het subcommando "Cn" kunt u de kleur aangeven waarin een lijn moet worden getekend; hierin is n het nummer van de kleur (zie Appendix C). Zo nodig kunt u elk lijnstuk een andere kleur geven door het desbetreffende subcommando vooraf te laten gaan door "Cn". De kleur blijft dan hetzelfde tot het volgende subcommando "Cn".

2.11.11 Subcommando voor tekenen onder een hoek

Met het subcommando "An" kunt u een tekening in stappen van 90° verdraaien. n moet een waarde tussen 0 en 3 hebben. Deze getallen hebben de volgende betekenis:

0 = 0°
1 = 90°
2 = 180°
3 = 270°



2.11.12 Subcommando voor vergroten of verkleinen

Het subcommando "Sn" is een schaalfactor, waarmee u een tekening kunt verkleinen of vergroten. n is een getal

tussen 1 en 255, dat door 4 wordt gedeeld om de schaalfactor te bepalen. De schaalfactor is het getal waarmee de lijnlengte, gedefinieerd met de subcommando's "U", "E", "R" enz. en "M" wordt vermenigvuldigd. Deze schaalfactor blijft van kracht totdat een andere schaalfactor wordt gedefinieerd, ook als u het programma onderbreekt en opnieuw start.
De instructie DRAW "S8R10" betekent bij voorbeeld dat een lijn naar rechts wordt getekend, niet met een lengte van 10 maar van 20 beeldpunten (namelijk 10 x 8 : 4) als gevolg van het subcommando "S8".

De computer start met de schaalfactor "S4" (= 4/4 = 1). "S0" (nul) heeft hetzelfde effect als "S4".

2.11.13 **Subcommando's als strings**

Het zal u zijn opgevallen dat alle subcommando's tussen aanhalingstekens staan en door MSX-BASIC dus als strings worden beschouwd. Dit betekent dat u in plaats van stringconstanten in de DRAW-instructie ook stringvariabelen kunt gebruiken.
Onderbreek het programma met de toetsen CTRL en STOP en tik het volgende programma in:

```
NEW
10 SCREEN 2
20 A$="E20R20G20L20"
30 DRAW "A0S4BM80,80XA$;"
40 GOTO 40
RUN
```

Dit programma tekent een ruit op het scherm. De noodzakelijke subcommando's zijn ondergebracht in de stringvariabele A$, die door de DRAW-instructie wordt uitgevoerd met gebruikmaking van het subcommando "X". Na de stringvariabele A$ moet een puntkomma (;) worden ingetoetst.
Het gebruik van variabelen in plaats van constanten is zinvol als een bepaalde string meermalen moet worden gebruikt, zoals in het volgende programma:

```
NEW
10 SCREEN 2
20 A$="E20R20G20L20"
```

```
30 DRAW "AOS4BM80,80XA$;"
35 DRAW "A1S8BM160,160XA$;"
40 GOTO 40
RUN
```

Alle constanten die als subcommando in een DRAW-instructie worden gebruikt, mogen worden vervangen door variabelen als de volgende formule wordt gebruikt: " = variabele-naam". Een voorbeeld:

```
NEW
10 SCREEN 2:PSET (80,80),15
20 X1=10:X2=5
30 DRAW "M+=X1;,-=X2;"
RUN
```

2.11.14 Samenvatting

De DRAW-instructie biedt vele mogelijkheden. Wij raden u aan al die mogelijkheden te proberen. Om u te helpen, hierbij nog een voorbeeld:

```
NEW
10 SCREEN 2: COLOR 15,4,4
20 PSET (127,95),1
30 FOR I=1 TO 189 STEP 4
40 A$ ="L"+STR$(I)+"D"+STR$(I+1)+"R
"+STR$(I+2)+"U"+STR$(I+3)
50 DRAW "S4XA$;"
60 NEXT I
70 CLS:GOTO 20
RUN
```

U kunt het programma zoals altijd onderbreken door tegelijk de toetsen CTRL en STOP in te drukken.

2.12 "Sprites"

2.12.1 Wat zijn "sprites" ?

"Sprites" zijn bewegende figuren op het scherm. Het woord laat zich het beste vertalen met "geesten", maar we zullen hier toch maar de Engelse naam gebruiken. Een kenmerkende eigenschap van sprites is dat ze zich min of meer vóór het scherm lijken te bewegen doordat de grafische afbeeldingen, die u met de instructies uit §

2.11 hebt gemaakt, intact blijven.
Om met sprites te kunnen werken, moet aan twee voorwaarden zijn voldaan:

1 De sprite moet zijn gedefinieerd. Met andere woorden: u moet aangeven hoe hij eruit moet zien.
2 U moet beslissen of de sprite zichtbaar zal zijn op het scherm.

2.12.2 Het definiëren van sprites

Sprites kunnen in twee formaten worden gedefinieerd: groot en klein. Kleine sprites zijn opgebouwd uit 8 x 8 beeldelementen (Engels: picture elements of kortweg pixels), ook wel beeldpunten genoemd. U kunt maximaal 256 kleine sprites definiëren. Dat definiëren gebeurt met behulp van de stringvariabele SPRITE$. Zie ook Appendix E.
Laten we aannemen dat u de volgende sprite wilt definiëren:

8	4	2	1	8	4	2	1	
								H 0 0
								H 0 0
								H 0 0
	■	■		■	■			H 6 C
■			■			■		H 9 2
			■					H 1 0
								H 0 0
								H 0 0

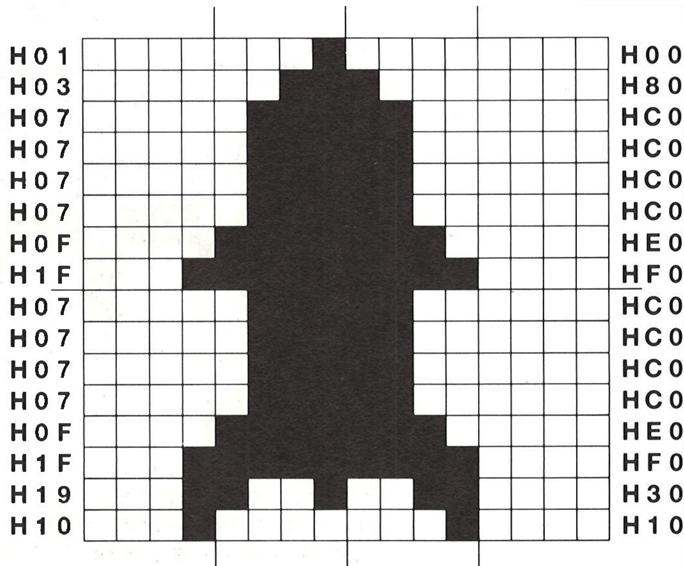
Elke horizontale rij beeldelementen vormt één byte, die is verdeeld in 2 x 4 bits. Elke groep van 4 bits (Engels: "nibble") krijgt een hexadecimale waarde, zoals is aangegeven aan de rechterkant van de afbeelding. Die waarde is afhankelijk van de beeldinhoud van de nibble en wordt berekend door de kolomwaarden van de beeldelementen, die boven de afbeelding staan, op te tellen.

Voor de linker-"nibble" in de vierde regel in de afbeelding geldt: 4 + 2 = 6 = &H6. Voor de rechter-"nibble" van dezelfde regel geldt: 8 + 4 = 12 = &HC.

De grootte van de sprite wordt aangegeven in de SCREEN-instructie. De sprite uit voorgaande afbeelding kan als volgt worden gecodeerd:

```
10 SCREEN 2,0
20 A$=CHR$(&H00)+CHR$(&H00)+CHR$(&
H00)+CHR$(&H6C)+CHR$(&H92)+CHR$(&H
10)+CHR$(&H00)+CHR$(&H00)
30 SPRITE$(1)=A$
```

U kunt ook grote sprites definiëren, opgebouwd uit 16 x 16 beeldpunten. U kunt maximaal 64 grote sprites definiëren.
Laten we aannemen dat u een grote sprite wilt definiëren:



Elke horizontale rij van 16 beeldpunten bestaat uit 2 bytes, die elk weer zijn verdeeld in 2 x 4 bits. Deze worden op dezelfde manier in hexadecimale vorm gecodeerd als bij de kleine sprites.
Grote sprites worden in verticale richting gecodeerd, zoals in het volgende voorbeeld:

```
10 SCREEN 2,2
20 A$=CHR$(&H01)+CHR$(&H03)+CHR$(&
H07)+CHR$(&H07)+CHR$(&H07)+CHR$(&H
07)+CHR$(&H0F)+CHR$(&H1F)
30 B$=CHR$(&H07)+CHR$(&H07)+CHR$(&
H07)+CHR$(&H07)+CHR$(&H0F)+CHR$(&H
1F)+CHR$(&H19)+CHR$(&H10)
40 C$=CHR$(&H00)+CHR$(&H80)+CHR$(&
HC0)+CHR$(&HC0)+CHR$(&HC0)+CHR$(&H
C0)+CHR$(&HE0)+CHR$(&HF0)
50 D$=CHR$(&HC0)+CHR$(&HC0)+CHR$(&
HC0)+CHR$(&HC0)+CHR$(&HE0)+CHR$(&H
F0)+CHR$(&H30)+CHR$(&H10)
60 SPRITE$(1)=A$+B$+C$+D$
```

2.12.3 Een kortere manier om sprites te definiëren

Het laatste programma is niet alleen een hoop tikwerk, men kan ook gemakkelijk vergissingen maken. Gelukkig is er een kortere manier om sprites te definiëren. Het volgende voorbeeld laat dat zien voor een kleine sprite:

```
10 SCREEN 2,0
20 DATA 0,0,0,108,146,16,0,0
30 A$=""
40 FOR I=1 TO 8
50 READ A:A$=A$+CHR$(A)
60 NEXT I
70 SPRITE$(1)=A$
```

In dit voorbeeld zijn de decimale waarden van de acht bytes opgenomen in een DATA-regel. Die waarden worden in regel 50 gelezen en toegekend aan stringvariabele A$. Als u geen zin hebt de hexadecimale waarden om te rekenen in decimale, kunt u dezelfde sprite ook als volgt definiëren:

```
20 DATA 0,0,0,6C,92,10,0,0
30 A$=""
40 FOR I=1 TO 8
50 READ B$:A$=A$+CHR$(VAL("&H"+B$))
60 NEXT I
70 SPRITE$(1)=A$
```

De grote sprite kan als volgt worden gedefinieerd:

```
10 SCREEN 2,0
20 DATA 1,3,7,7,7,7,15,31
30 DATA 7,7,7,7,15,31,25,16
40 DATA 0,128,192,192,192,192,224
,240
50 DATA 192,192,192,192,224,240,48
,16
60 A$=""
70 FOR I=1 TO 32
80 READ A:A$=A$+CHR$(A)
90 NEXT I
100 SPRITE$(1)=A$
```

In dit voorbeeld is de decimale waarde van de 32 bytes opgenomen in vier DATA-regels. In regel 80 worden deze waarden gelezen en toegevoegd aan string-variabele A$.
Net als bij de kleine sprite kunt u ook hier de hexadecimale waarden in de DATA-regels zetten en regel 80 vervangen door:

```
80 READ B$:A$=A$+CHR$(VAL("&H"+B$))
```

2.12.4 Sprites op het scherm plaatsen

Om sprites op het scherm te kunnen brengen moet onmiddellijk na de SCREEN-instructie, die het schermtype definieert, een code worden geplaatst die aangeeft welk type sprites zal worden gebruikt. Die code heeft de volgende betekenis:
0 = kleine sprites van 8 x 8 beeldpunten
1 = kleine sprites, vergroot tot 16 x 16 beeldpunten
2 = grote sprites van 16 x 16 beeldpunten
3 = grote sprites, vergroot tot 32 x 32 beeldpunten

De programmaregel: "10 SCREEN 2,3" betekent dat het schermtype 2 is (grafische stand 1) en het sprite-type 3 (grote sprites van 32 x 32 beeldelementen).

We hebben het nog niet met zoveel woorden gezegd, maar uit het voorgaande zal duidelijk zijn geworden dat u sprites kunt vergroten tot tweemaal het formaat dat u hebt gedefinieerd.

Er zijn enkele beperkingen aan het gebruik van sprites:
a Sprites kunnen niet worden gebruikt in tekststand 1 en 2.
b Het is niet mogelijk sprites van verschillende formaten door elkaar te gebruiken.
c Op een horizontale regel van het scherm kunnen tegelijkertijd ten hoogste vier sprites worden geplaatst.

Een sprite kan op het scherm worden geplaatst met de instructie PUT SPRITE, gevolgd door de volgende informatie:
a Rangnummer
b De positie op het scherm
c De kleur van de sprite
d Het nummer van de sprite

Het rangnummer is een getal van 0 tot en met 31, dat de volgende betekenis heeft: als twee sprites op dezelfde plaats op het scherm moeten komen heeft de sprite met het laagste rangnummer voorrang op de sprite met het hoogste nummer.

De positie op het scherm wordt weer aangegeven in X- en Y-coördinaten. De X-coördinaat is een getal van -32 tot 255 en de Y-coördinaat is een getal van -32 tot 191. Met deze waarden is het dus mogelijk een sprite buiten het scherm te plaatsen. De X- en Y-coördinaten hebben betrekking op het beeldpunt linksboven van de sprite (dus niet op het midden). Ook in dit geval mogen de coördinaten een absolute en een relatieve waarde hebben.
Als de Y-coördinaat een waarde heeft van 209 heeft, verdwijnt de sprite van het scherm.

De kleur van de sprite wordt aangegeven met de kleurcode (zie Appendix C).

Het nummer van de sprite bepaalt welke sprite op het scherm zal verschijnen.

Onderbreek het programma en toets het volgende voorbeeld in:

```
NEW
10 SCREEN 2,0
20 DATA 0,3,31,63,255,255,63,7
30 DATA 0,128,248,255,255,254,252
,192
40 DATA 0,0,0,108,146,16,0,0
50 FOR I=1 TO 3
60 A$=""
70 FOR J=1 TO 8
80 READ A:A$=A$+CHR$(A)
90 NEXT J
100 SPRITE$(I)=A$
110 NEXT I
120 PUT SPRITE 4,(180,50),15,1
130 PUT SPRITE 5,(188,50),15,2
140 FOR I=-32 TO 212
150 PUT SPRITE 3,(I,50),1,3
160 NEXT I
170 PUT SPRITE 3,(I,209)
180 FOR I=212 TO -32 STEP -1
190 PUT SPRITE 3,(I,50),1,3
200 NEXT I
210 GOTO 140
RUN
```

Als u dit programma foutloos hebt ingetoetst en het laat lopen, zult u een zwarte vogel op het scherm zien die door een witte wolk vliegt. De witte wolk is in twee sprites (nummer 1 en 2) gedefinieerd. De zwarte vogel is gedefinieerd als sprite nummer 3. De witte wolk wordt door de instructies in regel 120 en 130 op het scherm gebracht. De zwarte vogel beweegt van links naar rechts als gevolg van de FOR....NEXT-lus in regel 140 tot 160. De FOR...NEXT-lus in de regels 180 tot 200 maakt dat de vogel weer terugvliegt van rechts naar links.

Onderbreek nu het programma en verander regel 10 als volgt:

```
10 SCREEN 2,1
```

U ziet dat de sprites nu tweemaal zo groot zijn. Onderbreek het programma nogmaals en verander regel 190 als volgt:

```
190 PUT SPRITE 6,(I,50),1,3
```

Als u het programma nu weer laat lopen, zult u zien dat de vogel nog steeds van links naar rechts vliegt maar nu op de terugweg achter de wolk langs vliegt. Dat komt omdat hij nu op de terugweg een hoger rangnummer heeft dan de wolk in regel 130.

2.12.5 Botsingen tussen sprites

MSX-BASIC maakt het mogelijk een bepaalde subroutine uit te voeren als twee sprites op het scherm botsen. Dit gebeurt met de instructies SPRITE ON, waarmee de functie wordt geactiveerd die controleert of er een botsing tussen sprites plaats vindt, en ON SPRITE GOSUB. Onderbreek weer het programma, laat het in het geheugen staan en verander het door de volgende programmaregels in te typen:

```
5 SCREEN 2,0
10 ON SPRITE GOSUB 300
135 SPRITE ON
210 GOTO 135
300 PUT SPRITE 3,(I,209)
310 FOR J=50 TO 191
320 PUT SPRITE 3,(I,J),1,3
325 NEXT J
330 I=0:SPRITE OFF
340 RETURN 140
```

Laat het programma nu weer lopen. Regel 10 geeft aan naar welke subroutine MSX-BASIC moet springen als de twee sprites elkaar raken. In regel 135 wordt de botsfunctie geactiveerd. In de subroutine van de regels 300 tot 340 zorgt de FOR....NEXT-lus ervoor dat de vogel naar beneden valt. In regel 330 wordt de botsfunctie

weer uitgeschakeld en regel 340 maakt dat de computer na het uitvoeren van de subroutine terugkeert naar regel 140 van het hoofdprogramma.

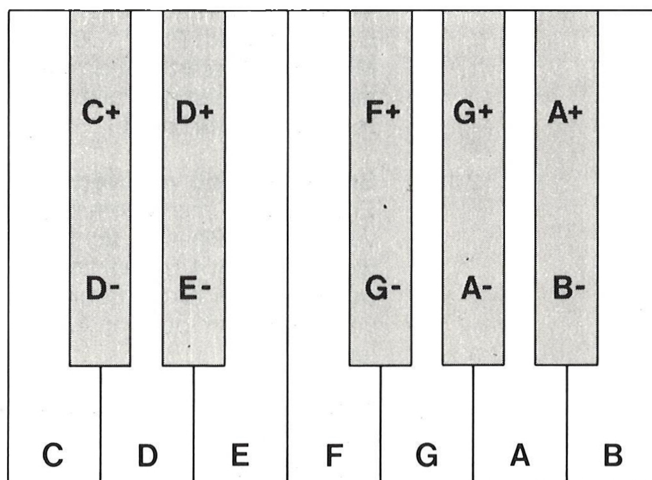
2.13 Muziek

2.13.1 Inleiding

MSX-BASIC kent een speciale instructie, PLAY, waarmee u uw computer muziek kunt laten maken. Door gebruik te maken van deze instructie kunt u driestemmige muziek programmeren. De muziek voor elk van deze stemmen wordt geprogrammeerd door middel van een aantal subcommando's, zoals de instructie PLAY.

2.13.2 Subcommando's voor de tonen van een octaaf

De letters "A" tot en met "G" worden gebruikt om aan te geven welke toon moet worden opgewekt. "A + " betekent dat A een halve toon wordt verhoogd tot ais. "A-" betekent verlaging met een halve toon tot as. MSX-BASIC werkt in feite als het manuaal van een piano. Dit betekent dat " + " alleen mag worden gebruikt als op de desbetreffende witte pianotoets een zwarte toets volgt. "-" mag alleen worden gebruikt als zich links van de corresponderende witte pianotoets een zwarte toets bevindt. Combinaties zoals "F-" en "E + " zijn dus niet toegestaan.
Een voorbeeld: PLAY "CEG".



2.13.3 **Subcommando voor de octaven**

De toonhoogte kan worden ingesteld met het subcommando "On", waarin O de eerste letter van het woord Octaaf is en n een getal tussen 1 en 8. De toonomvang van de computer is dus acht octaven.
Een standaard-octaaf, zoals op een piano, loopt van C tot en met B.
Een voorbeeld: PLAY "O5CEGO4CEGO5CEG".

2.13.4 **Subcommando voor de toonhoogte**

Een andere manier om te toonhoogte aan te geven is met het subcommando "Nn", waarin n een getal is tussen 0 en 96. De beide volgende voorbeelden hebben hetzelfde effect:

```
PLAY "O3CEG"
PLAY "N24N28N31"
```

2.13.5 **Subcommando voor verandering van de lengte van de noten**

Het subcommando "Ln" wordt gebruikt om de lengte van de noten aan te geven. In dit geval is n een getal tussen 1 en 64, dat de volgende betekenis heeft:
1 = een hele noot
2 = een halve noot
3 = een derde noot

4 = een kwart noot enz.
Een voorbeeld: PLAY "L8DEL4DCE".
MSX-BASIC start met de waarde 4. Als u het subcommando "Ln" niet gebruikt hebben de tonen dus de lengte van een kwart noot.

2.13.6 Subcommando voor verlenging van de noten

Er is nog een manier om de lengte van de noten te veranderen. Plaatst u achter een toon een punt (.), dan wordt de noot 1½ maal zo lang. U mag verscheidene punten achter elkaar gebruiken. PLAY "A..." geeft bij voorbeeld een toon van $1,5^3 = 3,375$ maal de normale lengte.

2.13.7 Subcommando voor het tempo

Het tempo kan worden ingesteld met het subcommando "Tn", waarin n een getal tussen 32 en 255 is. Het getal geeft het aantal kwartnoten per minuut aan.

2.13.8 Subcommando voor pauzes

Een pauze kunt u introduceren met het subcommando "Rn". Hierin is n een getal tussen 1 en 64. De lengte van de pauze bij een bepaalde waarde van n komt overeen met die van een toon met dezelfde waarde van n (zie § 2.13.4).

2.13.9 Subcommando voor volume

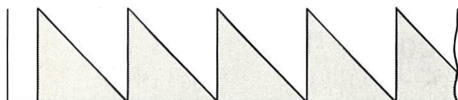
De geluidssterkte van een toon kan worden bepaald met het subcommando "Vn", waarin n een getal tussen 0 en 15 is. Het laatste getal komt overeen met het grootste volume.
Voorbeeld: PLAY "V15CGR1V9CG".

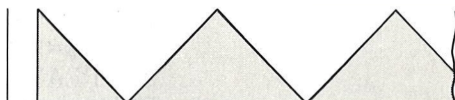
2.13.10 Subcommando voor de vorm van het geluid

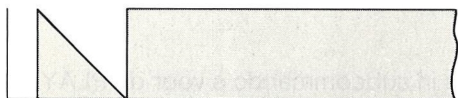
Het subcommando "Sn" wordt gebruikt om de vorm van de toon te bepalen, bij voorbeeld aanzwellend, afnemend, periodiek aanzwellend en afnemend enz. n is een getal tussen 0 en 15, dat overeenkomt met de volgende vormen:

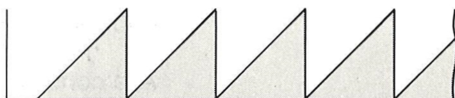

0...3 en 9

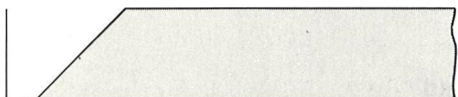

4...7 en 15


8


10


11

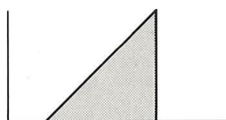

12


13


14

2.13.11 **Subcommando voor de lengte van de geluidsvorm**
"Mn" is het subcommando dat kan worden gebruikt om de lengte van de geluidsvorm aan te geven. n is een getal tussen 1 en 65535. Luister naar de volgende geluidseffecten:

```
PLAY "S15M500C"
PLAY "S15M1000C"
PLAY "S12M500C"
PLAY "S12M1000C"
```



Hoe kleiner de waarde van n is, des te korter duurt de geluidsvorm.

2.13.12 **Subcommando's met stringvariabelen**
MSX-BASIC maakt het mogelijk alle subcommando's op te nemen in een alfanumerieke variabele en in de PLAY-instructie deze stringvariabelen aan te roepen. Het laatste voorbeeld kan ook als volgt worden geprogrammeerd:

```
NEW
10 A$="S12M1000C"
20 PLAY "XA$;"
RUN
```

Direct achter de variabele moet een puntkomma worden geplaatst.

Alle constanten, die in subcommando's voor de PLAY-instructie worden gebruikt, kunnen worden vervangen door een variabele, en wel op de volgende manier: "=naam van de variabele;". Bij voorbeeld: PLAY "N=X1;".

2.13.13 Meerstemmig geluid

Alle subcommando's van de instructie PLAY mogen voor alle drie de stemmen worden gebruikt, mits een komma wordt gezet tussen de subcommando's voor elke stem. Bij voorbeeld: PLAY "O4A","O5C","O3B".

Uit het voorgaande blijkt wel dat de PLAY-instructie zeer veel mogelijkheden biedt. Die leert u alleen kennen door ze allemaal te proberen. Om u te helpen hierbij nog een voorbeeld:

```
NEW
5 CLEAR 1000
10 A$="V8S1M9000T250L4O5"
20 B$="F+F+8E2R8EE8D2R8R4F+F+F+G2
F+"
30 C$="O4BO5DDO4BO5D2EO4BO5D"
40 D$="O4B.O5DR4.B.A.DD1R2."
50 H$=B$+".R8"+B$+"O4A"+C$+"DDD8C+
2R4O4A"+C$+".D8D.D8E.R8R2"+B$+".R8
"+D$
60 I$="R1R1R4AAAB2A.R8R1R1R4AAAB2A
R1R1R1R1R1R1R1R1R1RR4R8AAAB2A"
70 PLAY "XA$;","XA$;"
80 PLAY "XH$;","XI$;"
RUN
```

Herkent u deze melodie ?
Let wel goed op het verschil tussen de hoofdletter O en de nul (0).

2.13.14 De functie PLAY

Met de functie PLAY (niet te verwarren met de instructie PLAY) kunt u controleren of de PLAY-instructie helemaal is uitgevoerd. U kunt dit proberen door de volgende regels toe te voegen aan het voorgaande programma:

```
85 CLS
90 PRINT PLAY(1):GOTO 90
```

Laat het programma opnieuw lopen. Terwijl de melodie wordt gespeeld, verschijnt de waarde -1 op het scherm. U kunt het programma weer onderbreken door tegelijkertijd de toetsen CTRL en STOP in te drukken.

2.13.15 De instructie SOUND

Met de instructie SOUND kan een bepaalde waarde in één van de registers van de geluidsgenerator worden gezet. Gebruik deze instructie alleen als u volkomen vertrouwd bent met de manier waarop deze generator werkt.

2.14 MSX-BASIC en de spelregelaars

2.14.1 Inleiding

Op uw MSX-computer kunnen twee spelregelaars ("joy sticks") worden aangesloten. U kunt MSX-BASIC laten reageren op de richting waarin de spelregelaars worden bewogen. Voor dat doel kunt u echter ook de cursortoetsen gebruiken.

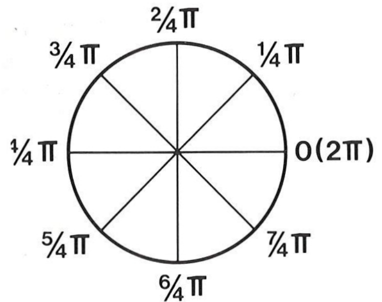
2.14.2 Uitlezen van de richting

De richting waarin de spelregelaar wordt bewogen kan worden gecontroleerd met de functie STICK(n). Hierin kan n de waarden 0, 1 en 2 hebben. Deze waarden hebben de volgende betekenis:

0 = De cursortoetsen worden gebruikt als spelregelaar.
1 = De spelregelaar verbonden met aansluiting 1.
2 = De spelregelaar eerbonden met aansluiting 2.

De waarden die worden verkregen met de functie STICK(n) liggen tussen 0 en 8. Die waarden hebben de volgende betekenis:

0 = neutraal
1 = noord
2 = noordoost
3 = oost
4 = zuidoost
5 = zuid
6 = zuidwest
7 = west
8 = noordwest



Een voorbeeld:

```
NEW
10 A=STICK(0)
20 IF A=STICK(0) THEN 20
30 PRINT STICK(0)
40 GOTO 10
RUN
```

In dit voorbeeld wordt de variabele A in regel 10 gelijk gemaakt aan de waarde van de ingedrukte cursortoets. Zolang geen cursortoets is ingedrukt, zal A de waarde nul hebben. Wordt één van de cursortoetsen ingedrukt, dan zal regel 30 worden uitgevoerd. Probeer alle cursortoetsen en stop daarna het programma met de toetsen CTRL en STOP.

2.14.3 De actietoets

MSX-BASIC kan ook onderzoeken of de actietoets van de spelregelaar is ingedrukt. In plaats van de actietoets mag u ook de spatiebalk van het toetsenbord gebruiken. Dat onderzoeken gebeurt met de functie STRIG(n), waarbij n een getal is tussen 0 en 4. Deze getallen hebben de volgende betekenis:

0 = De spatiebalk doet dienst als actietoets.
1 = De actietoets is toets 1 van de spelregelaar, verbonden met aansluiting 1.
2 = De actietoets is toets 1 van de spelregelaar, verbonden met aansluiting 2.
3 = De actietoets is toets 2 van de spelregelaar, verbonden met aansluiting 1.
4 = De actietoets is toets 2 van de spelregelaar, verbonden met aansluiting 2.

De waarde die de functie STRIG(n) oplevert is 0 als de actietoets niet is ingedrukt en -1 als de actietoets wel is ingedrukt. Een voorbeeld:

```
NEW
10 IF STRIG(0)=-1 THEN PRINT
STRIG(0)
20 GOTO 10
RUN
```

In regel 10 controleert MSX-BASIC of de spatiebalk is ingedrukt. Als dat het geval is, wordt de waarde van STRIG(0) op het scherm gezet.

2.14.4 Het lanceren van een raket

In het volgende voorbeeld wordt een raket gelanceerd als de spatiebalk wordt ingedrukt. Toets het programma zorgvuldig in en overtuig u ervan dat u geen fouten maakt.

```
NEW
10 SCREEN 2,3:COLOR 1,4,1
20 DATA 0,0,0,0,3,7,15,63
30 DATA 63,127,127,63,31,1,0,0
40 DATA 0,0,0,0,192,252,255,255
50 DATA 255,255,255,255,255,255,28
,0
60 DATA 0,0,0,1,15,31,255,255
70 DATA 255,255,255,255,255,255,30
,0
80 DATA 0,0,0,128,224,248,248,252
90 DATA 252,248,248,240,240,224,0,
0
100 DATA 0,0,0,28,126,127,255,127
110 DATA 31,15,7,3,0,0,0,0
120 DATA 0,0,0,0,96,240,248,252
130 DATA 254,254,252,252,248,0,0,0
140 DATA 0,0,0,0,0,0,3,15
150 DATA 63,255,255,127,63,31,0,0
160 DATA 0,0,0,0,0,248,252,254
170 DATA 254,252,252,252,252,248,2
48,0
180 DATA 0,0,0,0,0,0,0,0
190 DATA 0,0,0,0,0,0,0,0
```

```
200 DATA 3,3,7,7,7,7,7,15
210 DATA 15,15,15,31,31,31,31,31
220 DATA 0,0,0,0,0,0,0,0
230 DATA 0,0,0,0,0,0,0,0
240 DATA 255,255,255,16,40,16,0,0
250 DATA 0,0,0,0,0,0,0,0
260 DATA 255,255,255,0,1,0,0,0
270 DATA 0,0,0,0,0,0,0,0
280 DATA 240,240,240,128,64,128,0,
0
290 DATA 0,0,0,0,0,0,0,0
300 DATA 8,28,28,62,62,62,127,255
310 DATA 62,62,62,62,127,255,201,1
28
320 DATA 0,0,0,0,0,0,0,128
330 DATA 0,0,0,0,0,128,128,128
340 DATA 8,28,28,62,62,62,62,28
350 DATA 28,20,0,0,0,0,0,0
360 DATA 0,0,0,0,0,0,0,0
370 DATA 0,0,0,0,0,0,0,0
380 FOR J=1 TO 9:A$=""
390 FOR I=1 TO 32:READ A:A$=A$+CHR
$(A):NEXT I
400 SPRITE$(J)=A$:NEXT J
410 CIRCLE (88,120),16,6,-0.0001,-
3.415,0.5
420 PAINT (88,118),6
430 LINE (0,192)-(255,120),12,BF
440 PUT SPRITE 1,(80,16),15,1
450 PUT SPRITE 2,(112,16),15,2
460 PUT SPRITE 10,(160,80),15,3
470 PUT SPRITE 11,(16,48),15,4
480 I=120:J=144:GOSUB 1000
490 IF STRIG(0)=-1 GOTO 2000
500 IF STICK(0)=3 THEN GOSUB 3000
510 IF STICK(0)=7 THEN GOSUB 4000
520 GOTO 490
1000 PUT SPRITE 5,(I,J),1,5:J=J+32
1010 PUT SPRITE 6,(I,J),1,6:I=I+32
1020 PUT SPRITE 7,(I,J),1,7:J=J-32
1030 PUT SPRITE 8,(I,J),7,8:I=I-32
1040 RETURN
2000 I=I+32:FOR K=144 TO -33 STEP
-1
```

```
2010 PUT SPRITE 8,(I,K),7,8:M=K+32
2020 PUT SPRITE 9,(I,M),9,9
2025 FOR T=1 TO 1000:NEXT T
2030 PUT SPRITE 9,(I,209)
2040 NEXT K:FOR K=1 TO 1000:NEXT K
:COLOR 15,4,7:END
3000 I=I+1:IF I>200 THEN I=200
3010 GOSUB 1000:RETURN
4000 I=I-1:IF I<0 THEN I=0
4010 GOSUB 1000:RETURN
RUN
```

U kunt het lanceerplatform naar links en naar rechts bewegen met de cursortoetsen. De raket wordt gelanceerd als u de spatiebalk indrukt.
Als het programma niet goed werkt, controleer het dan zorgvuldig op tikfouten.

2.15 MSX-BASIC en de printer

Uw MSX-computer is uitgerust met een aansluiting voor een afdrukeenheid. Deze printer kunt u gebruiken om uw BASIC-programma's of de resultaten van die programma's op papier af te drukken.
U kunt een lijst van uw programma, dat in het geheugen van de computer staat, afdrukken met het commando LIST. Met hetzelfde commando kunt u ook een groter of kleiner deel van het programma afdrukken. Zie hiervoor het Naslagwerk voor Programmeurs.

De resultaten van een berekening of een bewerking afdrukken gaat met de instructie LPRINT. Een voorbeeld:

```
NEW
10 SCREEN 0
20 INPUT "Wat is uw naam";A$
30 LPRINT A$
40 END
RUN
```

In regel 20 vraagt dit programma uw naam in te tikken. De naam die u intikt wordt toegekend aan de variabele A$. In regel 30 wordt de inhoud van de variabele A$ afgedrukt door de printer.

2.15.1 De instructie OPEN

Er is nog een manier om gegevens naar de printer te sturen. U kunt de instructie OPEN gebruiken om een buffer te openen voor een randapparaat, bij voorbeeld een printer. Dit betekent dat u gegevens naar een randapparaat kunt sturen om die te laten verwerken. Hieronder een voorbeeld:

```
NEW
10 SCREEN 0
20 INPUT "Wat is uw naam";A$
30 OPEN "LPT:" AS #1
40 PRINT #1,A$
50 CLOSE #1
60 END
RUN
```

In regel 30 van dit voorbeeld wordt een buffer voor de printer geopend, met de instructie OPEN "LPT:". LPT is een afkorting van Line PrinTer, ofwel regeldrukker.
Met de instructie OPEN kunt u ook buffers voor andere randapparaten opnenen, bij voorbeeld:
OPEN "CAS:" opent een buffer voor een bestand op cassette.
OPEN "CRT:" opent een buffer voor een beeldscherm in de tekststand.
OPEN "GRP:" opent een buffer voor een beeldscherm in de grafische stand.
Na de instructie OPEN met het nummer van de buffer worden opgegeven met het nummerteken ("hekje"), bij voorbeeld #1. Elke keer dat u in uw programma naar dat nummer verwijst, weet MSX-BASIC welke buffer en welk randapparaat u bedoelt. De buffer wordt gesloten met de instructie CLOSE.

De instructie OPEN kan erg handig zijn, zoals uit het volgende voorbeeld blijkt:

```
NEW
10 SCREEN 0
20 ON KEY GOSUB 500,510,520,530,54
0,550
30 PRINT"Uitvoer naar:":PRINT
40 PRINT"F1 = cassette"
```

```
50 PRINT"F2 = printer"
60 PRINT"F3 = scherm tekststand 1
70 PRINT"F4 = scherm tekststand 2"
80 PRINT"F5 = scherm grafische sta
nd 1"
90 PRINT"F6 = scherm grafische sta
nd 2"
100 KEY(1) ON:KEY(2) ON:KEY(3) ON:
KEY(4) ON:KEY(5):KEY(6) ON
110 GOTO 110
500 OPEN"CAS:TEST" AS #1:GOTO 1000
510 OPEN"LPT:" AS #1:GOTO 1000
520 SCREEN 0:OPEN "CRT:" AS #1:GOT
O 1000
530 SCREEN 1:OPEN "CRT:" AS #1:GOT
O 1000
540 SCREEN 2:OPEN "CRT:" AS #1:GOT
O 1000
550 SCREEN 3:OPEN "CRT:" AS #1:GOT
O 1000
1000 KEY(1) OFF:KEY(2) OFF:KEY(3)
OFF:KEY(4) OFF:KEY(5) OFF:KEY(6) O
FF
1010 A$="MSX":PRINT #1,A$
1020 FOR I=1 TO 5000:NEXT
1030 CLOSE #1
1040 SCREEN 0
1050 RETURN 20
RUN
```

In dit voorbeeld kunt u kiezen uit zes "randapparaten" om de tekst "MSX" naar toe te sturnen. Probeer ze allemaal en zie wat er gebeurt.
Vergeet niet een lege cassette in uw datarecorder te steken en de toetsen PLAY en REC van de recorder in te drukken voordat u mogelijkheid 1 kiest.
U kunt het programma afbreken door tegelijk CTRL en STOP in te drukken.

3 ONDERHOUD

3.1 Onderhoud van de MSX-computer

Uw MSX-computer heeft vrijwel geen onderhoud nodig. Zo nodig kunt u hem afstoffen met een droge, niet pluizende doek. Gebruik nooit chemische middelen !

Als uw MSX-computer niet goed functioneert, als u vreemde geluiden hoort of een vreemde geur waarneemt, bij voorbeeld ten gevolge van oververhitting, schakel de computer dan onmiddellijk uit en uw handelaar in.
Reparaties mogen uitsluitend worden uitgevoerd door daartoe bevoegde dealers. Verwijder nooit zelf de kap van uw computer.

3.2 Onderhoud van de cassetterecorder

Als u een cassetterecorder gebruikt, verdient het aanbeveling van tijd tot tijd de koppen daarvan schoon te maken, bij voorbeeld met een Philips reinigingscassette 811/CCT. Zie hiervoor de gebruiksaanwijzing van de recorder.

Bewaar de cassettes op een koele en droge plaats. Vermijd direct zonlicht en hou ze uit de buurt van radiatoren en andere warmtebronnen en van magnetische velden. Doet u dat niet, dan kan de informatie op de cassetteband geheel of gedeeltelijk verloren gaan.

3.3 Dingen die u wel en niet moet doen

Zoals alle andere elektrische en elektronische apparaten is uw MSX-computer gevoelig voor vocht. Vermijd dus dat er vloeistof in de computer kan komen.

In de kast van de MSX-computer zijn ventilatieopeningen aangebracht. Hou deze openingen altijd vrij, zodat de lucht op de juiste manier door het apparaat kan circuleren.
Stel de computer niet bloot aan direct zonlicht of de warmte van radiatoren, kachels en andere warmtebronnen.

Raak nooit met uw vingers de contacten van de insteek-

modules, de aansluitstekers en dergelijke aan omdat ze dan kunnen gaan oxyderen.

Zorg ervoor dat niemand over de aansluitkabels van uw computer kan struikelen.

Moet u een steker losnemen, pak die dan stevig vast en trek hem behoedzaam los. Probeer nooit een steker los te nemen door aan de kabel te trekken.

Laat uw MSX-computer nooit vallen als u hem verplaatst, en laat ook nooit zware voorwerpen op de computer vallen.

4 APPENDICES

4.1 Inleiding

In dit hoofdstuk vindt u een tiental appendices (A tot en met J) met extra informatie over uw MSX-computer:

A Overzicht van foutmeldingen
B Wiskundige functies
C Kleurtabel
D Besturingsfuncties
E Tekenset
F Gereserveerde woorden
G Toetsenbord
H Technische gegevens
I Video Display Processor (VDP)
J Programmeerbare geluidsgenerator

4.2 Verantwoording

De bewerker van deze gebruiksaanwijzing heeft geprobeerd Engelse termen zo veel mogelijk te vermijden. Niettemin zult u vooral in deze appendices nogal wat Engelse woorden tegenkomen. Dat is onvermijdelijk omdat bij voorbeeld de foutmeldingen in het Engels op het scherm verschijnen. In deze gevallen zullen wij de Nederlandse vertaling erbij zetten.
In andere gevallen zijn Engelse termen gehandhaafd òf omdat er geen goed Nederlands woord voor bestaat, bij voorbeeld sprite, òf omdat het Engelse woord ook in het Nederlands is ingeburgerd, zoals cursor in plaats van het onhandelbare woord positieaanwijzer. (Overigens is "cursor" geen Engels maar een Latijns woord, dat "gerechtsbode" betekent.)

Bij de technische specificaties hebben wij soms de Engelse omschrijving laten staan. U hebt deze gegevens niet direct nodig om verdienstelijk te leren programmeren. De specificaties zijn meer bestemd voor de technici onder u, die zelf uitbreidingen willen bouwen. Vertalen van deze termen heeft vaak alleen maar tot gevolg dat ook de doorgewinterde technicus er niets meer van begrijpt. Verversingscyclussignaal zegt hem niet veel; Refresh cycle signal is meteen duidelijk.

Een bijzonder geval is het Engelse woord "file", dat

wordt gebruikt voor alles dat op een cassetteband of een diskette kan worden weggeschreven. Daarbij wordt dus niet altijd onderscheid gemaakt tussen programma's, bestanden en dergelijke, zoals wij bij voorkeur doen.

Appendix A Overzicht van foutmeldingen

Foutmelding	Code	Verklaring
Bad file name	56	(Foute programma- of bestandsnaam) U hebt een verkeerde naam gebruikt om een programma of een bestand aan te duiden.
Bad file number	52	(Fout bestandsnummer) Het nummer dat u gebruikt verwijst naar een bestand dat nog niet geopend is met de instructie OPEN, of het nummer is hoger dan het aantal bestanden dat met MAXFILES is gedefinieerd.
Can't CONTINUE	17	(Kan niet doorgaan) U probeert, door CONT in te typen, door te gaan met een programma dat: ● onderbroken is door een foutmelding ● gewijzigd is nadat de uitvoering is afgebroken ● niet bestaat.
Device I/O error	19	(Invoer- of uitvoerfout) Tijdens het inlezen of wegschrijven van een bestand of een programma is een fout geconstateerd.
Direct statement	57	(Directe opdracht) Tijdens het inlezen van een ASCII-bestand is een directe opdracht gevonden.
Division by zero	11	(Delen door nul) Bij het uitvoeren van een berekening ontdekt de computer een deling door 0 of een machtsverheffing van 0 met een negatieve exponent. De nul kan ook het resultaat zijn van een voorgaande berekening.
Field overflow	50	(Veld loopt over) Het aantal bytes, toegekend met de instructie FIELD, is groter dan 256.
File already open	54	(Bestand al geopend) U vraagt de computer met de instructie

		OPEN een bestand te openen dat al geopend is, of u probeert met KILL een bestand te wissen dat geopend is.
File not open	59	(Bestand niet geopend) U probeert een bestand in te lezen of weg te schrijven dat nog niet is geopend met de instructie OPEN.
Illegal direct	12	(Niet toegestane directe opdracht) U hebt een instructie ingetoetst die in de directe stand niet is toegestaan.
Illegal function call	5	(Niet toegestane functie-aanroep) Bij het aanroepen van een functie is een verkeerde parameter (waarde) meegegeven. Dit kan het geval zijn bij: ● een negatieve of te grote index, bij voorbeeld A(-2). ● Nul of een negatieve waarde bij een LOG-functie. ● Een negatieve waarde bij een SQR-functie. ● Een onjuiste waarde bij één van de volgende instructies: MID$, LEFT$, RIGHT$, INP, OUT, PEEK, POKE, TAB, SPC, STRING$, SPACE$, DELETE, INSTR$, ON....GOTO of ON....GOSUB
Input past end	55	(Invoer voorbij einde) U probeert gegevens te lezen uit een bestand dat al helemaal gelezen is. Gebruik de functie EOF om deze fout te voorkomen.
Internal error	51	(Interne fout) Er doet zich een onverwachte situatie voor die niet kan worden opgelost door de BASIC-interpreter.
Line buffer overflow	25	(Regelbuffer loopt over) U hebt een te lange regel ingetoetst.

Missing operand	24	(Operand ontbreekt) Een expressie bevat een functie, een instructie of een commando zonder operand, dat wil zeggen zonder getal of string waarop de bewerking kan worden uitgevoerd.
NEXT without FOR	1	(NEXT zonder FOR) De variabele in de NEXT-instructie correspondeert niet met de variabele in de laatste FOR-instructie, of de computer komt een NEXT-instructie tegen zonder voorafgaande FOR-instructie.
No RESUME	21	(Geen RESUME) Na het afhandelen van een fout ontbreekt de instructie RESUME (= hervat het programma).
Out of DATA	4	(Geen DATA meer) Er wordt een READ-instructie uitgevoerd terwijl er geen gegevens meer zijn die nog gelezen kunnen worden.
Out of memory	7	(Geen geheugen meer) Een programma is te groot of bevat te veel FOR-lussen, GOSUB-instructies of variabelen. Het kan ook te ingewikkelde expressies bevatten.
Out of string space	14	(Geen stringruimte meer) Er zijn meer alfanumerieke variabelen dan de gereserveerde stringruimte kan bevatten. Gebruik de instructie CLEAR om meer stringruimte te reserveren.
Overflow	6	(Register loopt over) Het resultaat van een berekening is groter dan bij de gekozen variabele kan worden weergegeven.

Redimensioned array	10	(Array opnieuw gedimensioneerd) Er zijn twee DIM-instructies voor dezelfde variabele gegeven, of een variabele die al in gebruik is wordt geDIMensioneerd.
RESUME without error	22	(RESUME zonder foutmelding) De computer wordt met een RESUME-instructie gevraagd het programma te hervatten zonder dat de instructie ON ERROR GOTO is uitgevoerd.
RETURN without GOSUB	3	(RETURN zonder GOSUB) De computer komt de instructie RETURN tegen zonder dat er een GOSUB aan vooraf is gegaan.
Sequential I/O only	58	(Alleen sequentiële in- en uitvoer) U probeert in een sequentieel bestand te lezen of te schrijven alsof het een "random"-bestand (willekeurig toegankelijk bestand) is.
String formula too complex	16	(Stringexpressie te complex) De alfanumerieke uitdrukking is te lang of te ingewikkeld.
String too long	15	(String te lang) Er is geprobeerd meer dan 255 alfanumerieke tekens in een string te stoppen.
Subscript out of range	9	(Index buiten het bereik) Er wordt een poging gedaan een element te gebruiken met een index buiten het geDIMensioneerde gebied.
Syntax error	2	(Syntaxis-fout) Een commando, instructie of functie bevat een spelfout, een onjuiste interpunctie of zondigt tegen de taalregels van MSX-BASIC.

Type mismatch	13	(Verkeerd type) Aan een numerieke variabele wordt een alfanumerieke waarde toegekend of omgekeerd, of aan een functie is een alfanumerieke waarde toegekend in plaats van een numerieke waarde.
Unidentified line number	8	(Onbekend regelnummer) Een commando of een instructie verwijst naar een niet bestaande programmaregel.
Unidentified user function	18	(Onbekende gebruikersfunctie) Er wordt een gebruikersfunctie aangeroepen voordat die is gedefinieerd met de instructie DEF.
Unprintable error	23 26...49 60...255	(Foutmelding zonder omschrijving) Deze foutcodes zijn in MSX-BASIC niet gedefinieerd. U mag ze gebruiken om uw eigen foutmeldingen te definiëren.
Verify error	20	(Verificatiefout) Deze foutmelding treedt op als het programma in het geheugen verschilt met het programma op cassette na het uitvoeren van de instructie CLOAD?.

Appendix B Wiskundige functies

Een aantal wiskundige functies kan niet rechtstreeks door MSX-BASIC worden uitgevoerd. Niettemin kunt u deze functies uitvoeren door gebruik te maken van de hieronder gegeven afgeleide functies. Moet een niet in MSX-BASIC beschikbare wiskundige functie in een programma meermalen worden uitgevoerd, dan is het zinvol deze functie aan het begin van het programma te definiëren. Daarna kan die functie telkens worden aangeroepen als dat nodig is.

Een voorbeeld:

```
NEW
10 DEF FN SEC(X) = 1/COS(X)
20 INPUT X
30 PRINT COS(X)
40 PRINT FN SEC(X)
50 GOTO 20
RUN
```

Afgeleide functie	Definitie MSX-BASIC
Cotangens	= 1/TAN(X)
Secans	= 1/COS(X)
Cosecans	= 1/SIN(X)
Inverse sinus	= ATN(X/SQR(-X*X+1))
Inverse cosinus	= -ATN(X/SQR(-X*X+1))+1.5708
Inverse cotangens	= ATN(X)+1.5708
Inverse secans	= ATN(X/SQR(X*X-1))+SGN(SGN(X)-1)*1.5708
Inverse cosecans	= ATN(X/SQR(X*X-1))+(SGN(X)-1)*1.5708
Sinus hyperbolicus	= (EXP(X)-EXP(-X))/2
Cosinus hyperbolicus	= (EXP(X)+EXP(-X))/2
Tangens hyperbolicus	= EXP(-X)/(EXP(X)+EXP(-X))*2+1
Cotangens hyperbolicus	= EXP(-X)/(EXP(X)-EXP(-X))*2+1
Secans hyperbolicus	= 2/(EXP(X)+EXP(-X))

Cosecans hyperbolicus	= 2/(EXP(X)-EXP(-X))
Inverse sinus hyperbolicus	= LOG(X + SQR(X*X + 1))
Inverse cosinus hyperbolicus	= LOG(X + SQR(X*X-1))
Inverse tangens hyperbolicus	= LOG((1 + X)/(1-X))/2
Inverse cotangens hyperbolicus	= LOG((X + 1)/(X-1))/2
Inverse secans hyperbolicus	= LOG((SQR(-X*X + 1) + 1)/X)
Inverse cosecans hyperbolicus	= LOG((SGN(X)*SQR(X*X + 1) + 1)/X)

Appendix C Kleurtabel

Kleurcode	Kleur
0	Transparant
1	Zwart
2	Groen
3	Lichtgroen
4	Donkerblauw
5	Lichtblauw
6	Donkerrood
7	Cyaan
8	Rood
9	Lichtrood
10	Donkergeel
11	Lichtgeel
12	Donkergroen
13	Magenta (paars)
14	Grijs
15	Wit

Appendix D Besturingsfuncties

De volgende tabel geeft een overzicht van alle besturingsfuncties van MSX-BASIC. Deze functies worden aangeroepen door tegelijkertijd CTRL en een andere toets, vermeld in de tweede kolom, in te drukken. Voor een aantal functies beschikt uw MSX-computer over speciale toetsen. Deze staan in de derde kolom. U kunt bij voorbeeld tabuleren door het indrukken van CTRL + I en door TAB in te drukken.
Elke besturingsfunctie heeft een code. Deze code kunt u in uw programma's gebruiken, in combinatie met de instructie CHR$(X).
Tabuleren kan bij voorbeeld worden geprogrammeerd als PRINT CHR$(9).
Een ander voorbeeld: PRINT CHR$(7) geeft een pieptoon.

Een beschrijving van de besturingsfuncties uit onderstaande tabel vindt u op de volgende pagina's.

CODE	**CTRL + toets:**	**Speciale toets**	**Besturingsfunctie**
1	A		Alternatieve tekens
2	B		Cursor naar voorgaande woord
3	C		Stop AUTO-commando
4	D		-
5	E		Wis rest van de regel
6	F		Cursor naar volgende woord
11	K	HOME	Cursor naar linker-bovenhoek
12	L	CLR	Wis het scherm (Clear)
13	M	RETURN	Cursor naar begin volgende regel
14	N		Cursor naar einde van de regel
15	O		-
16	P		-
17	Q		-
18	R	INS	Voeg in (Insert)
19	S		-
20	T		-
21	U		Wis regel
22	V		-
23	W		-
24	X	SELECT	-
25	Y		-

26	Z		-
27	[	ESC	-
28	\	→	Cursor naar rechts
29	]	←	Cursor naar links
30	↑	↑	Cursor omhoog
31	_	↓	Cursor omlaag

Beschrijving van de besturingsfuncties

Alternatieve tekens — Met deze functie wordt het volgende teken een alternatief teken (zie Appendix E).

Cursor naar voorgaande woord — De cursor gaat naar het eerste teken van het voorgaande woord. Voor MSX-BASIC is het eerste teken A...Z, a...z of 0...9.

Stop AUTO-commando — Met deze functie stopt MSX-BASIC met het genereren van regelnummers en keert terug naar de directe stand.

Wis rest van de regel — Alle tekens tot het eind van de regel worden gewist, het teken onder de cursor inbegrepen.

Cursor naar volgende woord — De cursor gaat naar het eerste teken van het volgende woord. Voor MSX-BASIC is het eerste teken A...Z, a...z of 0...9.

Pieptoon — Uit de luidspreker klinkt een pieptoon.

Wis voorgaande teken — Deze functie wist het teken links van de cursor. De cursor zelf en alle tekens rechts van de cursor gaan één positie naar links.

Tabuleer — De cursor gaat naar de eerstvolgende tabulatorstop. Er is een tabulatorstop op elke achtste positie.

Volgende regel — De cursor gaat naar de eerste positie op de volgende regel.

Cursor naar linker-bovenhoek — De cursor gaat naar de eerste positie van de eerste regel zonder het scherm te wissen.

Wis het scherm	De cursor gaat naar de eerste positie van de eerste regel en wist het scherm.
Cursor naar begin volgende regel	De logische regel (die één of meer schermregels kan omvatten) wordt doorgegeven aan MSX-BASIC (ENTER bij sommige computers) en de cursor gaat naar het begin van de volgende regel.
Cursor naar einde van de regel	De cursor gaat naar de positie onmiddellijk rechts van het laatste teken van de regel.
Voeg in	Met deze functie kunnen nieuwe tekens tussen de bestaande tekens op het scherm worden gevoegd. Bij nogmaals uitvoeren van de functie worden bestaande tekens overschreven.
Wis regel	De hele logische regel wordt gewist.
Cursor naar rechts	De cursor gaat één positie naar rechts.
Cursor naar links	De cursor gaat één positie naar links.
Cursor omhoog	De cursor gaat één regel omhoog.
Cursor omlaag	De cursor gaat één regel omlaag.

Appendix E Tekenset

Deze appendix geeft een overzicht van alle tekens die MSX-BASIC op het scherm kan zetten. Elk teken is opgebouwd uit een matrix van 8 x 8 beeldpunten.
Onder elk teken staan twee codes, links de decimale en rechts de hexadecimale code. Deze kunnen worden gebruikt in combinatie met de functie CHR$(X), waarin X de decimale of de hexadecimale code is. Er zijn verscheidene manieren om de hoofdletter A op het scherm te brengen, onder andere:

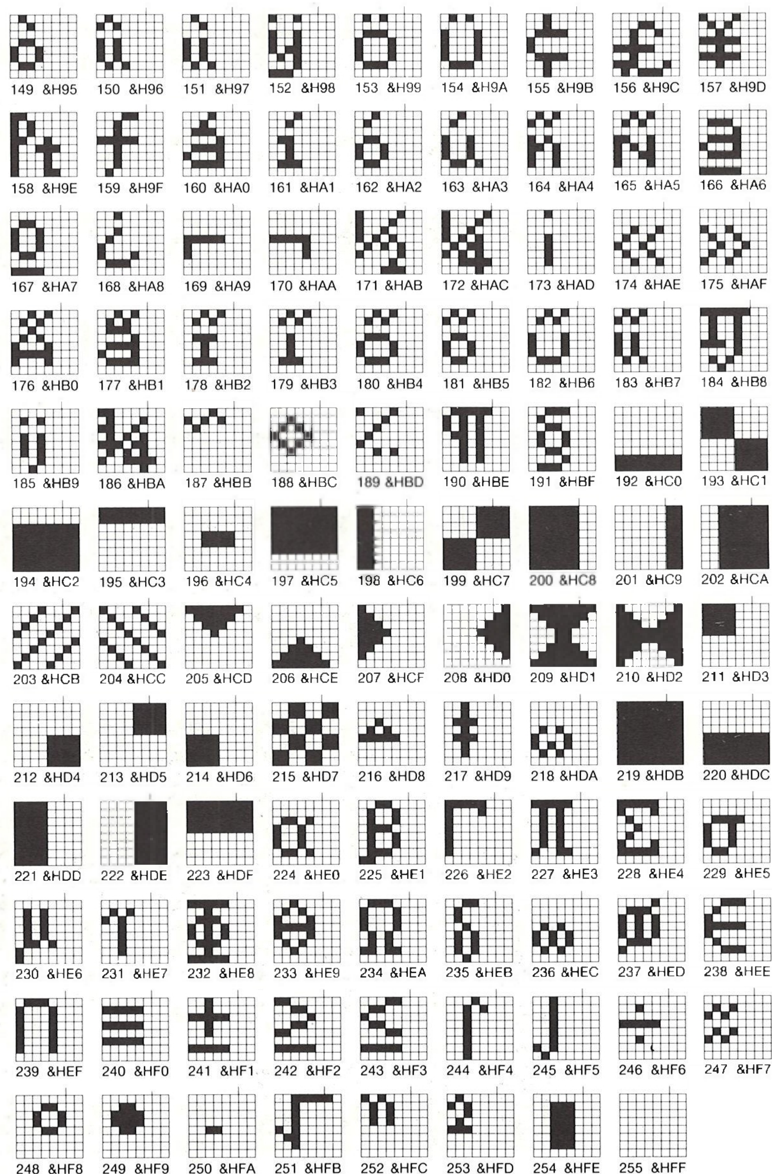
```
PRINT "A"
PRINT CHR$(65)
PRINT CHR$(&H41)
```

In tekststand 1, met ten hoogste 40 posities per regel, worden de laatste twee kolommen aan de rechterkant van elke matrix niet op het scherm gebracht. Bij de meeste tekens zijn de beeldpunten alleen in de eerste vijf kolommen geprogrammeerd. In tekststand 1 komen deze tekens dus dichter op elkaar te staan, met één kolom "wit" tussen de tekens. In tekststand 2 (SCREEN 1) worden alle acht kolommen op het scherm gebracht en is de afstand tussen de tekens meestal drie kolommen.
Bij sommige tekens, vooral bij grafische symbolen, zijn in de laatste drie kolommen wel beeldpunten geprogrammeerd. In tekststand 1 vallen de beeldpunten van de laatste twee kolommen dus weg. U kunt dit proberen met het volgende voorbeeld:

```
NEW
10 SCREEN 0
20 PRINT CHR$(210)
RUN
```

Verander nu regel 10 als volgt: 10 SCREEN 1. Als u het programma opnieuw laat lopen, zult u het verschil zien.

32 &H20	33 &H21	34 &H22	35 &H23	36 &H24	37 &H25	38 &H26	39 &H27	40 &H28
41 &H29	42 &H2A	43 &H2B	44 &H2C	45 &H2D	46 &H2E	47 &H2F	48 &H30	49 &H31
50 &H32	51 &H33	52 &H34	53 &H35	54 &H36	55 &H37	56 &H38	57 &H39	58 &H3A
59 &H3B	60 &H3C	61 &H3D	62 &H3E	63 &H3F	64 &H40	65 &H41	66 &H42	67 &H43
68 &H44	69 &H45	70 &H46	71 &H47	72 &H48	73 &H49	74 &H4A	75 &H4B	76 &H4C
77 &H4D	78 &H4E	79 &H4F	80 &H50	81 &H51	82 &H52	83 &H53	84 &H54	85 &H55
86 &H56	87 &H57	88 &H58	89 &H59	90 &H5A	91 &H5B	92 &H5C	93 &H5D	94 &H5E
95 &H5F	96 &H60	97 &H61	98 &H62	99 &H63	100 &H64	101 &H65	102 &H66	103 &H67
104 &H68	105 &H69	106 &H6A	107 &H6B	108 &H6C	109 &H6D	110 &H6E	111 &H6F	112 &H70
113 &H71	114 &H72	115 &H73	116 &H74	117 &H75	118 &H76	119 &H77	120 &H78	121 &H79
122 &H7A	123 &H7B	124 &H7C	125 &H7D	126 &H7E	127 &H7F	128 &H80	129 &H81	130 &H82
131 &H83	132 &H84	133 &H85	134 &H86	135 &H87	136 &H88	137 &H89	138 &H8A	139 &H8B
140 &H8C	141 &H8D	142 &H8E	143 &H8F	144 &H90	145 &H91	146 &H92	147 &H93	148 &H94



149 &H95
150 &H96
151 &H97
152 &H98
153 &H99
154 &H9A
155 &H9B
156 &H9C
157 &H9D
158 &H9E
159 &H9F
160 &HA0
161 &HA1
162 &HA2
163 &HA3
164 &HA4
165 &HA5
166 &HA6
167 &HA7
168 &HA8
169 &HA9
170 &HAA
171 &HAB
172 &HAC
173 &HAD
174 &HAE
175 &HAF
176 &HB0
177 &HB1
178 &HB2
179 &HB3
180 &HB4
181 &HB5
182 &HB6
183 &HB7
184 &HB8
185 &HB9
186 &HBA
187 &HBB
188 &HBC
189 &HBD
190 &HBE
191 &HBF
192 &HC0
193 &HC1
194 &HC2
195 &HC3
196 &HC4
197 &HC5
198 &HC6
199 &HC7
200 &HC8
201 &HC9
202 &HCA
203 &HCB
204 &HCC
205 &HCD
206 &HCE
207 &HCF
208 &HD0
209 &HD1
210 &HD2
211 &HD3
212 &HD4
213 &HD5
214 &HD6
215 &HD7
216 &HD8
217 &HD9
218 &HDA
219 &HDB
220 &HDC
221 &HDD
222 &HDE
223 &HDF
224 &HE0
225 &HE1
226 &HE2
227 &HE3
228 &HE4
229 &HE5
230 &HE6
231 &HE7
232 &HE8
233 &HE9
234 &HEA
235 &HEB
236 &HEC
237 &HED
238 &HEE
239 &HEF
240 &HF0
241 &HF1
242 &HF2
243 &HF3
244 &HF4
245 &HF5
246 &HF6
247 &HF7
248 &HF8
249 &HF9
250 &HFA
251 &HFB
252 &HFC
253 &HFD
254 &HFE
255 &HFF

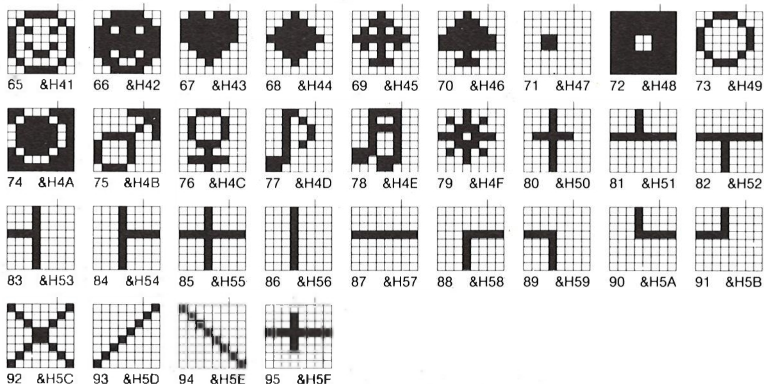
Alternatieve tekens

Behalve de 224 tekens van de vorige twee pagina's kent MSX-BASIC nog 31 alternatieve tekens met de codes 65...95. In een programma kunt u deze tekens oproepen door gebruik te maken van de instructie CHR$(1). Een voorbeeld:

```
NEW
10 SCREEN 1
20 PRINT CHR$(65)
30 PRINT CHR$(1)CHR$(65)
RUN
```

Regel 20 zet het normale teken op het scherm dat bij code 65 hoort, dat wil zeggen de hoofdletter A. Regel 30 zet een gezichtje ("Smile") op het scherm.

CHR$(1) moet worden herhaald voor elk alternatief teken dat u wilt oproepen. Hebt u veel alternatieve tekens nodig, bij voorbeeld om kaderlijnen te tekenen (codes 81...91), dan kunt u het tikwerk beperken door de codes onder te brengen in DATA-regels, zoals we in § 2.12 voor de sprites hebben gedaan.



Tekenmatrix

Elk teken van de tekenset is opgebouwd uit een matrix van 8 x 8 beeldpunten. Elke regel van de matrix is verdeeld in 2 x 4 beeldpunten, die respectievelijk de waarden 8, 4, 2 en 1 hebben. Als alle beeldpunten "gevuld" zijn, is de totale waarde van de vier blokjes: 8 + 4 + 2 + 1 = 15 = &HF. Zijn alle beeldpunten "leeg", dan is de totale waarde 0. In onderstaande afbeelding zijn alle mogelijkheden getekend om een rij van vier beeldpunten te vullen. Achter elke rij staat de decimale waarde die daarmee correspondeert, en voor de waarden 10...15 de hexadecimale waarde tussen haakjes.

De opbouw van de tekens gebeurt op dezelfde manier als bij de kleine sprites (zie § 2.12).

8 4 2 1

= 0

= 1

= 2

= 3

= 4

= 5

= 6

= 7

8 4 2 1

= 8

= 9

= 10 (A)

= 11 (B)

= 12 (C)

= 13 (D)

= 14 (E)

= 15 (F)

Appendix F Gereserveerde woorden

De volgende woorden zijn gereserveerd voor MSX-BASIC en mogen niet als namen voor variabelen worden gebruikt:

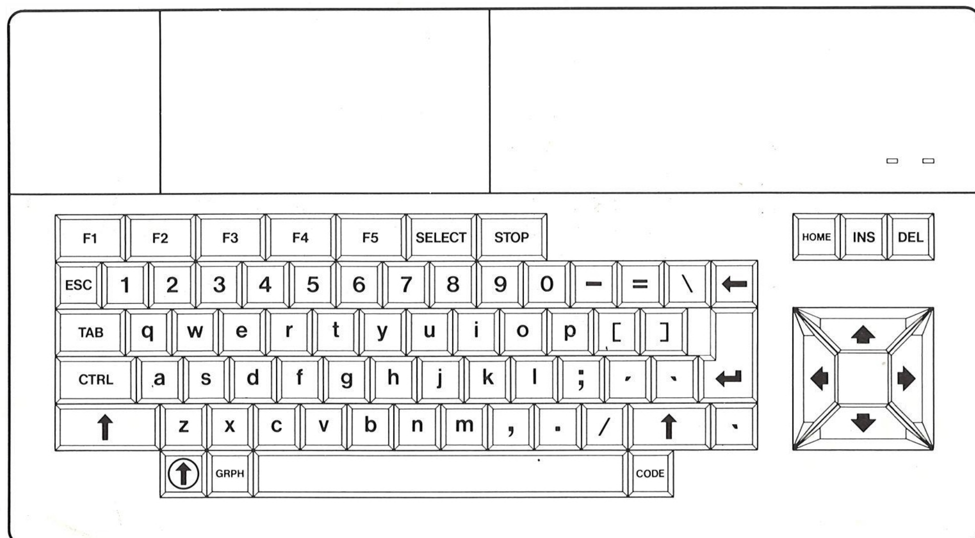
ABS
AND
AS
ASC
ATN
AUTO
BASE
BEEP
BIN$
BLOAD
BSAVE
CALL
CDBL
CHR$
CINT
CIRCLE
CLEAR
CLOAD
CLOSE
CLS
COLOR
CONT
COPY
COS
CSAVE
CSNG
CSRLIN
CVD
CVI
CVS
DATA
DEF
DEFINT
DEFDBL
DEFSNG
DEFSTR
DEFUSR
DELETE
DIM
DRAW
DSKF
ELSE
END
EOF
EQV
ERASE
ERL
ERR
ERROR
EXP
FIELD
FILES
FIX
FN
FOR
FRE
GET
GOSUB
GOTO
HEX$
IF
IMP
INKEY$
INP
INPUT
INPUT$
INSTR
INT
INTERVAL
KEY
KILL
LEFT$
LEN
LET
LINE
LIST
LLIST
LOAD
LOC
LOCATE
LOF
LOG
LPOS
LPRINT
LSET
MAXFILES
MERGE
MID$
MKD$
MKI$
MOD
MOTOR
MKS$
NAME
NEW
NEXT
NOT
OCT$
OFF
ON
OPEN
OR
OUT
PAD
PAINT
PDL
PEEK
PLAY
POINT
POKE
POS
PRINT
PSET
PRESET
PUT
READ
REM
RENUM
RESTORE
RESUME
RETURN
RIGHT$
RND
RSET
RUN
SAVE
SCREEN
SGN
SIN
SOUND
SPACE$
SPC
SPRITE
SPRITE$
SQR
STEP
STICK
STOP
STR$
STRIG
STRING$
SWAP
TAB
TAN
THEN
TIME
TROFF
TRON
USING
USR
VAL
VARPTR
VDP
VPEEK
VPOKE
WAIT
WIDTH
XOR

Appendix G Toetsenbord

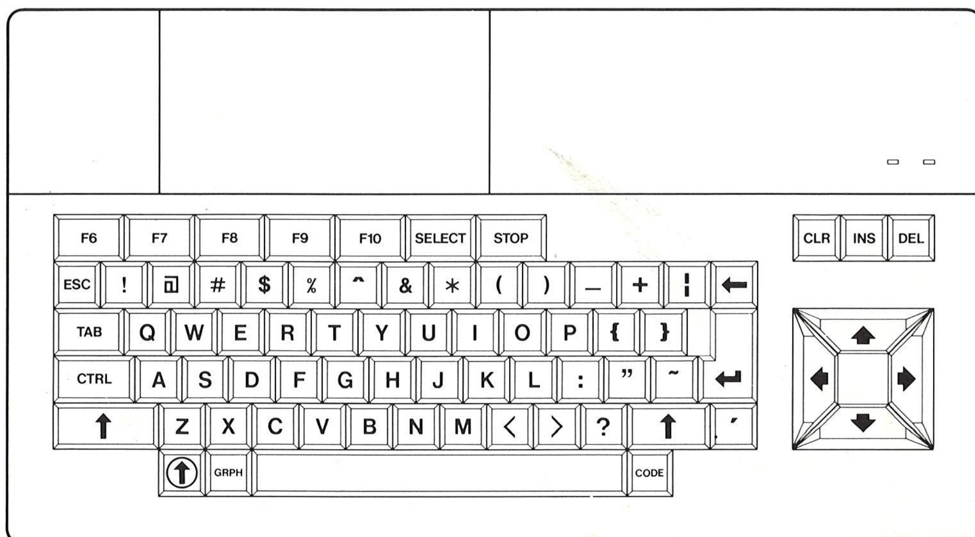
In deze appendix vindt u een overzicht van alle tekens die via het toetsenbord kunnen worden opgeroepen. Uw MSX-computer heeft in feite zes verschillende toetsenborden in één: normaal met of zonder SHIFT, GRPH met of zonder SHIFT en CODE met of zonder SHIFT.

Goed beschouwd is er nog een zevende toetsenbord, dat beschikbaar is nadat de CAPS-toets eenmaal is ingedrukt. In plaats van kleine letters produceert de computer dan hoofdletters, maar voor de andere tekens waarvoor u normaal de hoofdlettertoets moet indrukken, zoals die boven de cijfers, moet SHIFT worden gebruikt. Dit zevende toetsenbord noemt men wel het programmeer- of BASIC-toetsenbord.

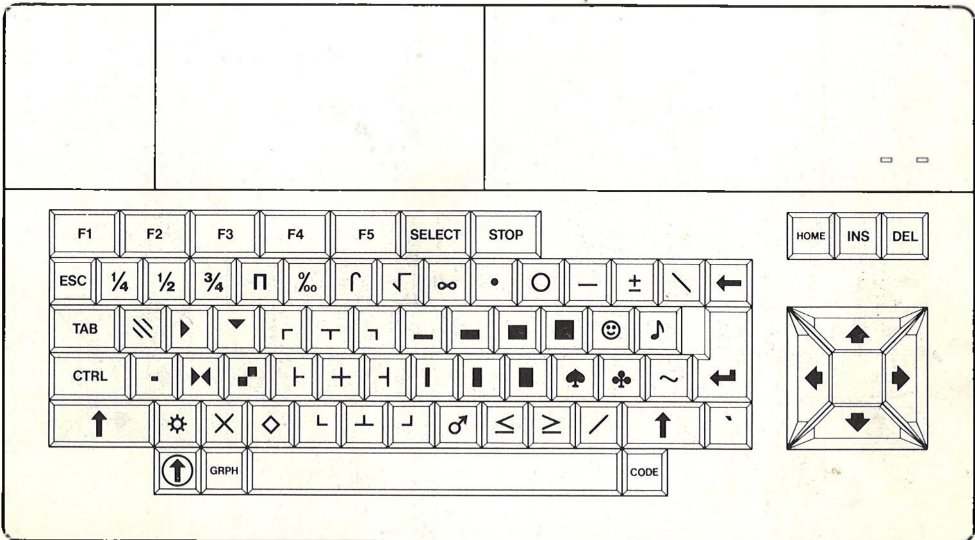
A Tekens bij het inschakelen van de computer, zonder dat SHIFT, CAPS, CODE of GRPH is ingedrukt.



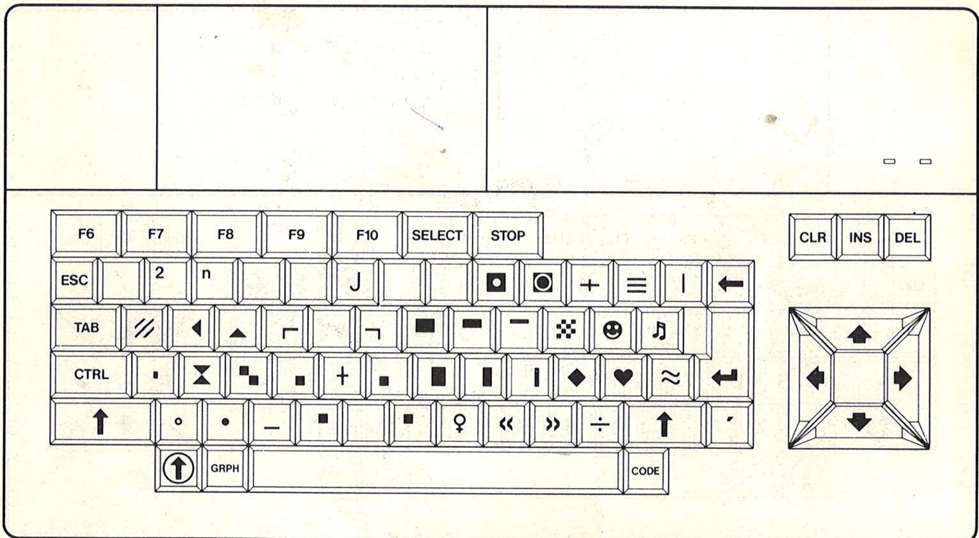
B Beschikbare tekens als een toets tegelijk met de hoofdlettertoets (SHIFT) wordt ingedrukt.



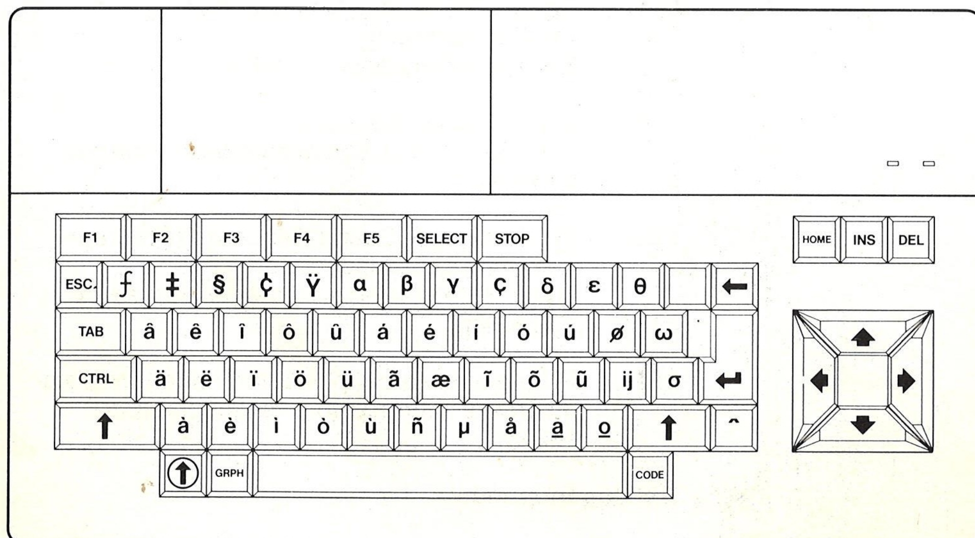
C Beschikbare tekens als de lettertoets tegelijk met de GRPH-toets wordt ingedrukt.



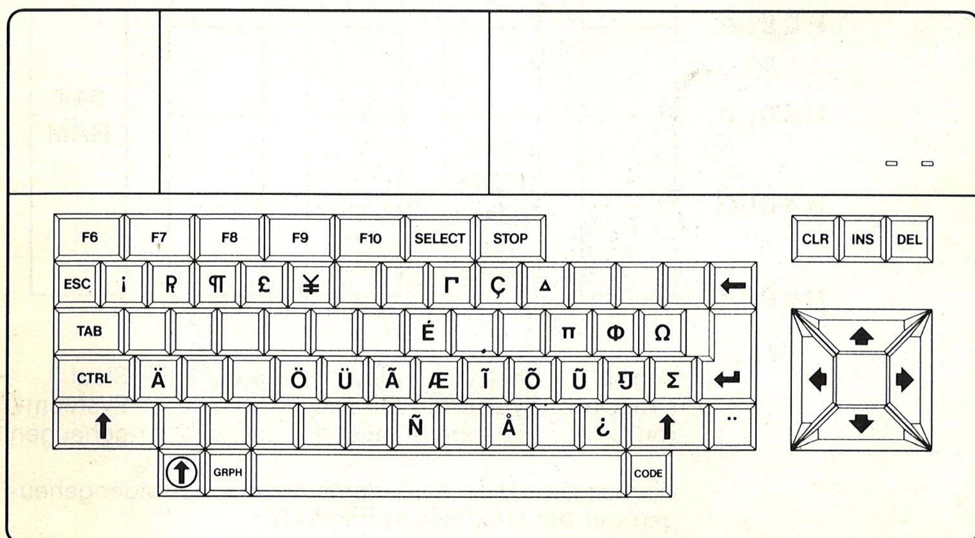
D Beschikbare tekens als de lettertoets tegelijk met de GRPH-toets en de SHIFT-toets wordt ingedrukt.



E Beschikbare tekens als de lettertoets tegelijk met de CODE-toets wordt ingedrukt.



F Beschikbare tekens als de lettertoets tegelijk met de CODE-toets en de SHIFT-toets wordt ingedrukt.



Appendix H Technische gegevens

1 Chip set

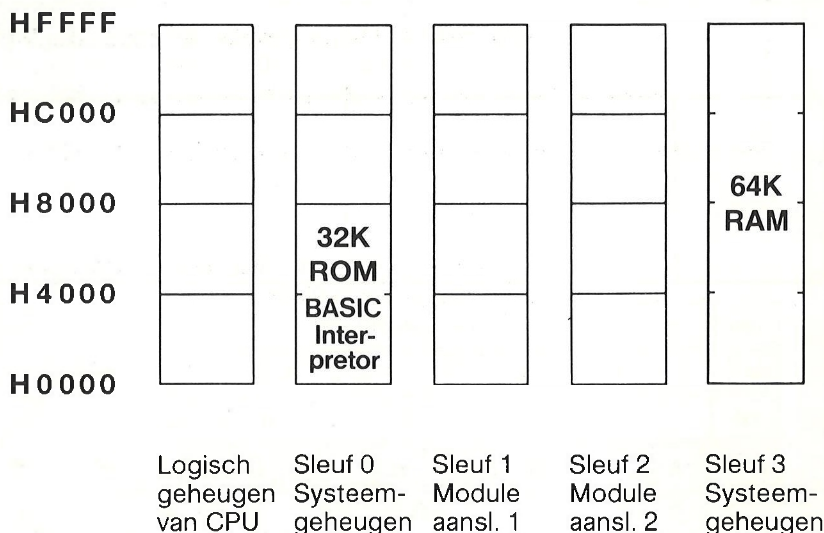
CPU Central Processing Unit (Centrale verwerkingseenheid):
Z80A, klokfrequentie 3,5 MHz.

VDP Video Display Processor:
TI TMS-9929A of equivalente geïntegreerde schakeling.

PSG Programmable Sound Generator (Programmeerbare geluidsgenerator):
GI AY-3-8910 of equivalente geïntegreerde schakeling.

PPI Programmable Peripheral Interface (Programmeerbare interface voor randapparatuur):
I 8255

2 Geheugenopbouw



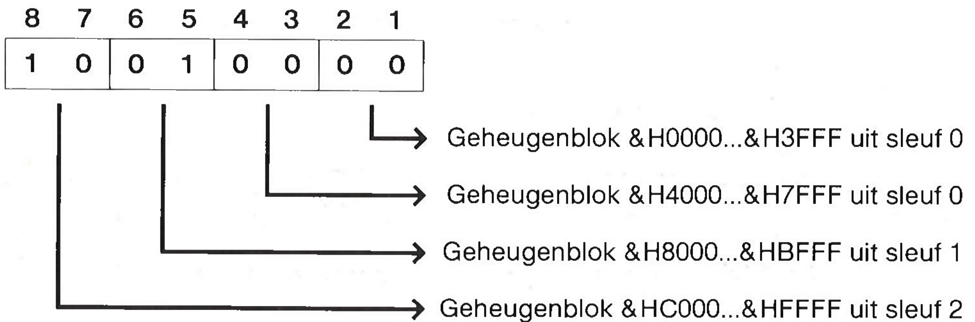
Logisch geheugen van CPU | Sleuf 0 Systeemgeheugen | Sleuf 1 Module aansl. 1 | Sleuf 2 Module aansl. 2 | Sleuf 3 Systeemgeheugen

Hiernaast heeft uw MSX-computer nog een videogeheugen van 16K RAM (zie Appendix I).

Bij gebruik van MSX-BASIC wordt de BASIC-interpreter (32 Kbyte) van sleuf 0 gekopieerd in het werkgeheugen (RAM) van sleuf 3. Daardoor blijft een vrij werkgeheugen van circa 32 Kbyte over voor programma's, gegevens en dergelijke.

De computer gebruikt de geheugenadressen &H8000...&H831F en &HF380...&HFFFF als werkruimte.

De logische geheugenruimte van de processor wordt gevuld met de fysieke geheugenruimte van de vier "sleuven". De manier waarop dat gebeurt wordt vastgelegd in het "slot select register", dat kan worden gevonden op poort A van de PPI. Bij wijze van voorbeeld:



Het adres van poort A kan worden gevonden in de input/output adrestabel.

3 Input/output adrestabel

Met MSX-BASIC is het mogelijk rechtstreeks te communiceren met de in- en uitgangspoorten door gebruik te maken van de instructies OUT en WAIT en de functie INP. Zie hiervoor het Naslagwerk voor Programmeurs.

HFF	
HB0	
	PPI
HA8	
	PSG
HA0	
	VDP
H98	
	PI
H90	
H00	

ADRESS	R/W	DESCRIPTION	REMARK
HA8	W	Port A data write	8255
	R	Port A data read	PPI
HA9	W	Port B data write	
	R	Port B data read	
HAA	W	Port C data write	
	R	Port C data read	
HAB	W	Mode set	
HA0	W	Adress latch	AY-3-8910
HA1	W	Data write	PSG
HA2	R	Data read	
H98	W	V-RAM Data write	TMS 9929A
	R	V-RAM Data read	VDP
H99	W	Command, adress set	
	R	Status read	
H90	R	Busy status (B1)	Printer
	W	Strobe output (B0)	Interface
H91	W	Print data	

4 Aansluitingen connectors voor insteekmodules

Pin 49 ... 1

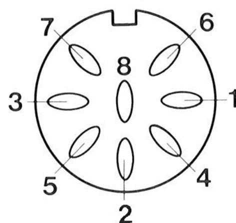
Pin 50 ... 2

Pin	Name	I/O	Pin	Name	I/O
1	CSI	O	2	CS2	O
3	CS12	O	4	SLTSL	O
5	Reserve	–	6	RFSH	O
7	WAIT	I	8	INT	I
9	M1	O	10	BUSDIR	I
11	IORQ	O	12	MERQ	O
13	WR	O	14	RD	O
15	RESET	O	16	Reserve	–
17	A9	O	18	A15	O
19	A11	O	20	A10	O
21	A7	O	22	A6	O
23	A12	O	24	A8	O
25	A14	O	26	A13	O
27	A1	O	28	A0	O
29	A3	O	30	A2	O
31	A5	O	32	A4	O
33	D1	I/O	34	D0	I/O
35	D3	I/O	36	D2	I/O
37	D5	I/O	38	D4	I/O
39	D7	I/O	40	D6	I/O
41	GND	–	42	CLOCK	O
43	GND	–	44	SW1	–
45	+ 5V	–	46	SW2	–
47	+ 5V	–	48	+ 12V	–
49	SOUNDIN	I	50	– 12V	–

Pin	Name	Content
1	CS1	ROM addresses 4000 ~ 7FFF select signal
2	CS2	ROM addresses 8000 ~ BFFF select signal
3	CS12	ROM addresses 4000 ~ BFFF select signal (for 256k ROM)
4	SLTSL	Slot select signal
5	Reserve	Reserved signal line ~ use inhibited
6	RFSH	Refresh cycle signal
7	WAIT	CPU's WAIT request signal
8	INT	Interrupt request signal to CPU
9	M1	Signal expressing CPU fetch cycle
10	BUSDIR	This signal controls direction of external data bus buffer Cartridges are selected and L level is output from each cartridge at data transmission time
11	IORQ	I/O request signal
12	MERQ	Memory request signal
13	WR	Write timing signal
14	RD	Read timing signal
15	RESET	System reset signal
16	Reserve	Reserved signal line ~ use inhibited
17 ~ 32	A0 ~ A15	Address bus signals
33 ~ 40	D0 ~ D7	Data bus signals
41	GND	Signal ground
42	CLOCK	CPU clock 3.579545MHz
43	GND	Signal ground
44, 46	SW1, SW2	For insertion/removal protect
45,47	+ 5V	+ 5V power source
48	+ 12V	+ 12V power source
49	SOUNDIN	Sound input signal (– 5bdm)
50	– 12V	– 12V power source

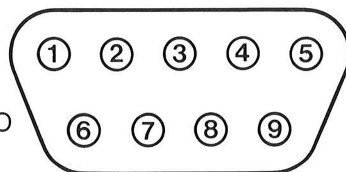
5 Aansluitingen datarecorder

Pen	Naam	I/O
1	Massa	
2	Massa	
3	Massa	
4	Signaal uit	O
5	Signaal in	I
6	Afstandb. +	O
7	Afstandb. -	O
8	Massa	



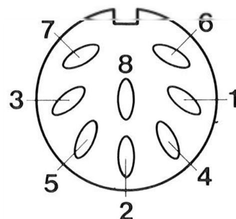
6 Aansluitingen spelregelaars ("joy sticks")

Pen	Naam	I/O
1	Omhoog	I
2	Omlaag	I
3	Links	I
4	Rechts	I
5	+5 V	
6	Trigger 1	I/O
7	Trigger 2	O
8	Signaal uit	O
9	Massa	



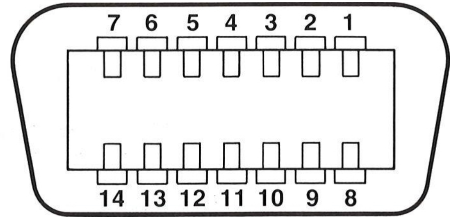
7 Aansluitingen monitor

Pen	Naam
1	+5 V
2	Massa
3	Geluid
4	Luminantie
5	Video
6	+12 V
7	
8	



8 Aansluitingen printer

Pen	Naam
1	PSTB
2	PDB0
3	PDB1
4	PDB2
5	PDB3
6	PDB4
7	PDB5
8	PDB6
9	PDB7
10	NC*
11	Busy
12	NC*
13	NC*
14	GND



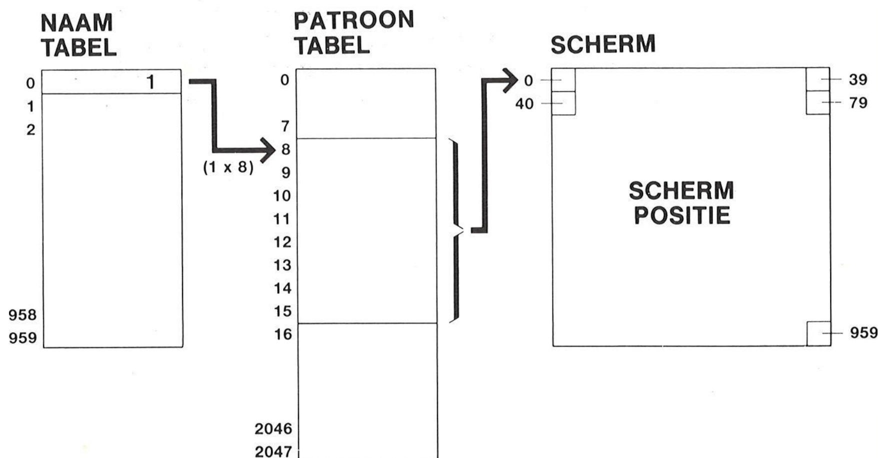
* NC = No Connection = niet aangesloten

Appendix I Video Display Processor (VDP)

De MSX-computer is uitgerust met een VDP. Deze levert alle informatie die nodig is om een beeld op het scherm te brengen. Daartoe beschikt hij over een eigen video-werkgeheugen (RAM) van 16 Kbyte.
Het videogeheugen kan worden uitgelezen met de functie VPEEK en er kan informatie rechtstreeks in dat geheugen worden geschreven met de instructie VPOKE.
Alle PRINT-instructies en grafische instructies zoals DRAW worden door MSX-BASIC automatisch verwerkt en in het videogeheugen gezet. Alles dat op uw scherm verschijnt is in het videogeheugen opgeslagen in de vorm van tabellen. Deze worden op verschillende manieren gebruikt, afhankelijk van het schermtype ("Screen mode").

1 Tekststand 1

In deze stand is het scherm verdeeld in 24 regels van elk 40 posities. In dit geval wordt gebruik gemaakt van twee tabellen in het videogeheugen: een naamtabel en een patroontabel. De naamtabel bevat de codes ("namen") van alle tekens die op het scherm moeten verschijnen. Voor elke naam zijn in de patroontabel 8 bytes gereserveerd, die bepalen hoe het teken eruit ziet (zie § 2.12 om een indruk te krijgen van de manier waarop de tekens worden opgebouwd). De plaats van een code in de naamtabel bepaalt de positie waar het teken op het scherm zal komen.



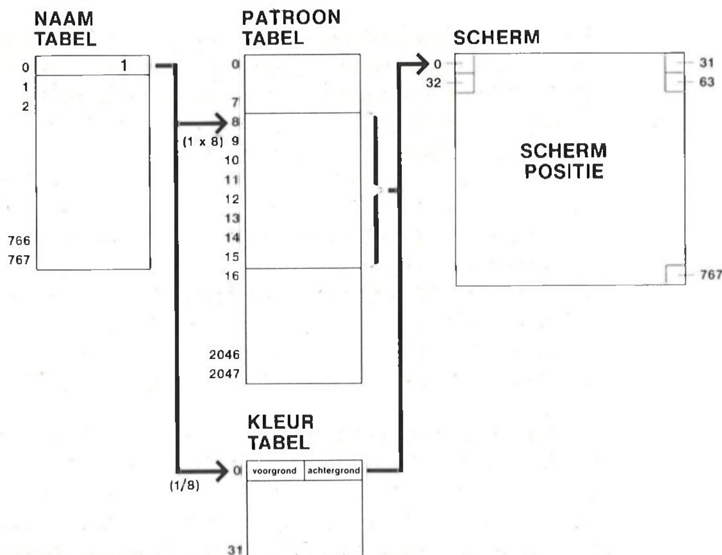
De plaats van de tabellen in het videogeheugen wordt aangegeven door de variabele BASE(n). Met PRINT BASE(0) kunt u het eerste adres van de naamtabel opvragen; PRINT BASE(2) geeft het eerste adres van de patroontabel. Eerst een voorbeeld.

```
NEW
10 SCREEN 0:WIDTH 40
20 PRINT "Hallo"
30 PRINT BASE(0),BASE(2)
40 A=VPEEK(BASE(0)):PRINT A
50 B=A*8:FOR I=B TO B+7
60 C$="00000000"+BIN$(VPEEK(BASE(2)
+I))
65 PRINT RIGHT$(C$,8)
70 NEXT I
RUN
```

In dit voorbeeld worden in regel 30 de beginadressen van de naamtabel en de patroontabel op het scherm gezet. De tekencode ("naam") van de letter "H" (de eerste letter van Hallo) wordt afgedrukt in regel 40. Vervolgens wordt het patroon van de letter "H" in binaire vorm gepresenteerd. Vergelijk dit patroon met dat van de letter "H" in Appendix E.

2 Tekststand 2

In deze stand is het scherm verdeeld in 24 regels van elk 32 posities. Er zijn nu drie tabellen in gebruik: een naam- en een patroontabel, zoals in tekststand 1, en verder een kleurtabel. De kleurtabel bevat 32 verschillende combinaties van voor- en achtergrondkleur.



De tekencodes 0 tot en met 7 van de naamtabel gebruiken de eerste kleurcombinatie uit de kleurtabel. De volgende 8 codes gebruiken de tweede kleurcombinatie enz. Een kleurcombinatie uit de kleurtabel wordt opgeslagen in de vorm van een byte, verdeeld in 2 x 4 bits. De eerste vier bits geven de voorgrondkleur aan en de tweede vier bits de achtergrondkleur.
De positie van de naamtabel wordt nu vastgelegd in de variabele BASE(5), de positie van de patroontabel in BASE(7) en de positie van de kleurtabel in BASE(6).
We kunnen nu het laatste programma als volgt veranderen:

```
NEW
10 SCREEN 1:WIDTH 32
20 PRINT "Hallo"
30 PRINT BASE(5),BASE(6),BASE(7)
40 A=VPEEK(BASE(5)):PRINT A
50 B=A*8:FOR I=B TO B+7
60 C$="00000000"+BIN$(VPEEK(BASE(7)
+I))
65 PRINT RIGHT$(C$,8)
70 NEXT I
```

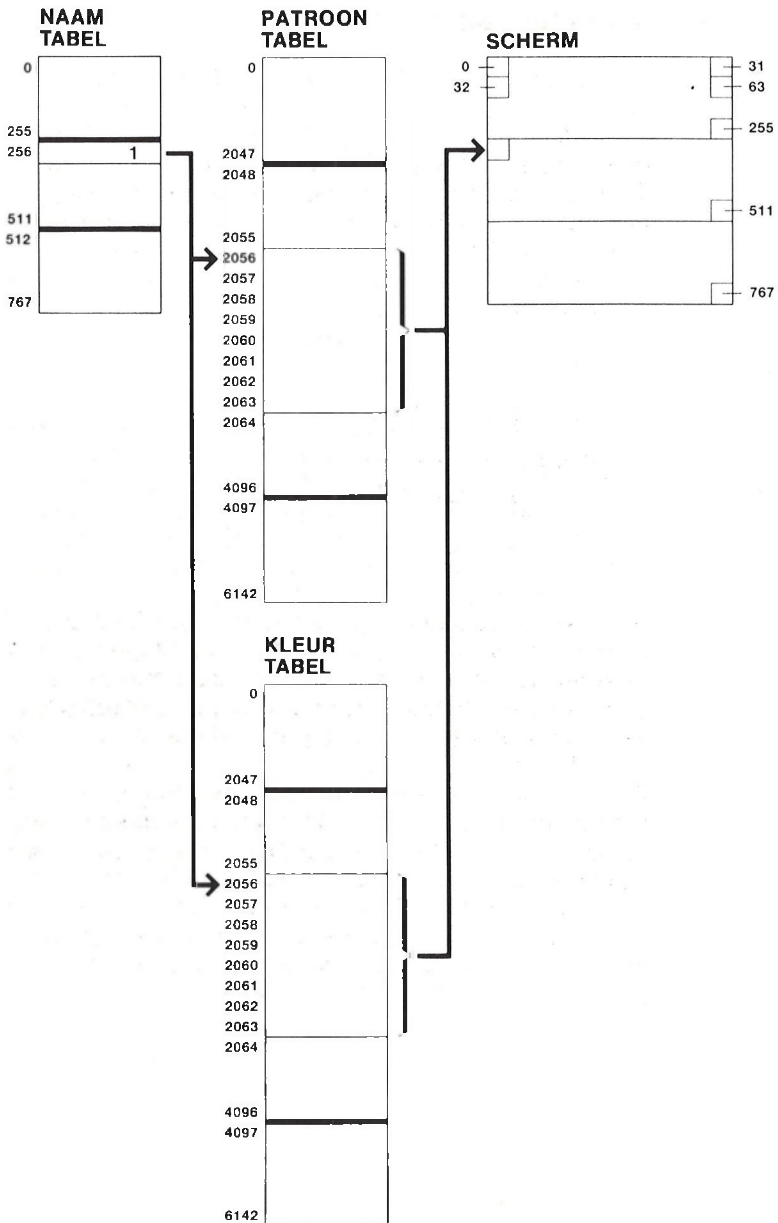
```
80 D=VPEEK(BASE(6))
90 PRINT USING "##";D/16
100 PRINT USING "##";D MOD 16
RUN
```

In dit voorbeeld worden de eerste adressen van naam-, patroon- en kleurtabel op het scherm gezet, gevolgd door de vorm en de naam van het teken (in dit geval de hoofdletter "H").
Hebt u dit voorbeeld bestudeerd, typ dan in: SCREEN 0:WIDTH 40.

3 Grafische stand 1

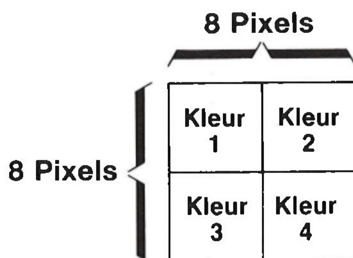
In deze stand is het scherm verdeeld in 3 x 8 regels met elk 32 posities. Ook in deze stand is er een naamtabel met codes. De plaats in de naamtabel correspondeert met een bepaalde positie op het scherm. De codes in de naamtabel geven aan waar het patroon en de kleur van het teken kunnen worden gevonden. Voor elke code in de naamtabel kan een patroon worden gedefinieerd in de patroontabel, die in dit geval 6144 bytes lang is (768 namen x 8 bytes). Voor elke byte in de patroontabel kunnen in de kleurtabel een voorgrond- en een achtergrondkleur worden gedefinieerd. Daarom heeft de kleurtabel dezelfde lengte als de patroontabel.

De plaats van de naamtabel in het videogeheugen is vastgelegd in de variabele BASE(10), de plaats van de kleurtabel in BASE(11) en die van de patroontabel in BASE(12).

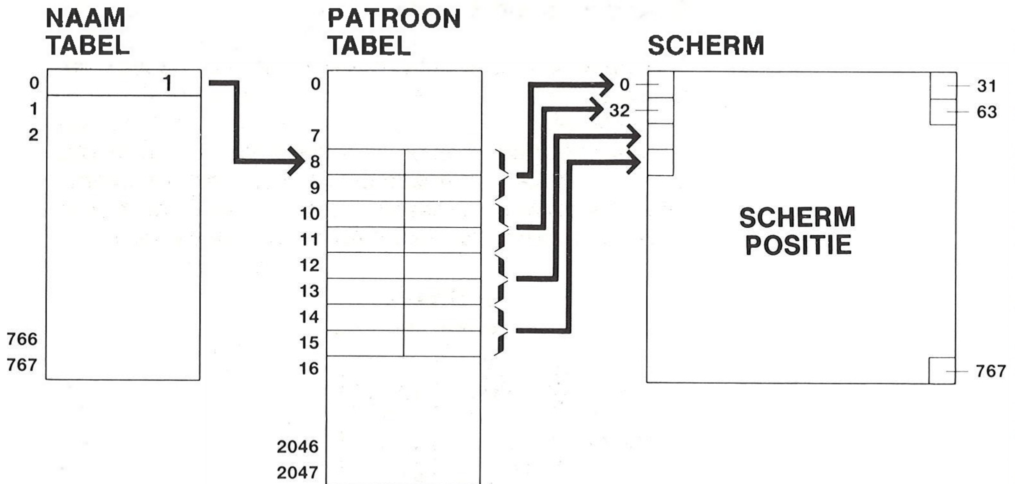

NAAM
TABEL
0
255
256
1
511
512
767
PATROON
TABEL
0
2047
2048
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
4096
4097
6142
SCHERM
0
32
31
63
255
511
767
KLEUR
TABEL
0
2047
2048
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
4096
4097
6142

4 Grafische stand 2

In deze stand is het scherm verdeeld in 24 regels van 32 posities. Elke positie bestaat uit vier blokjes van 4 x 4 beeldelementen ("pixels") en elk blokje kan een bepaalde kleur hebben. Ook bij dit schermtype wordt de positie op het scherm bepaald door de plaats in de naamtabel. De code in de naamtabel geeft aan waar de kleuren in de patroontabel kunnen worden gevonden.



Elke byte in de patroontabel is verdeeld in 2 x 4 bits; elke combinatie van 4 bits ("nibble") geeft het nummer van een kleur aan. Elke code in de naamtabel komt overeen met 8 bytes in de patroontabel. Slechts twee van deze acht bytes worden gebruikt om het beeld op het scherm te definiëren.
De eerste twee bytes van de acht die een groep vormen, worden gebruikt voor de namen van de bovenste regel. De volgende regel gebruikt de bytes 3 en 4, de daaropvolgende de bytes 5 en 6 en de laatste de bytes 7 en 8. Deze serie wordt herhaald voor de rest van het scherm. De positie van de naamtabel in het videogeheugen is vastgelegd in de variabele BASE(15), die van de patroontabel in de variabele BASE(17).



5 **Sprites**

Ook alle sprites worden in het videogeheugen geplaatst. De definitie van een sprite, dat wil zeggen hoe de sprite eruit ziet, is vastgelegd in de sprite-patroontabel. De plaats van de sprite op het scherm is opgeslagen in de sprite-attributietabel ("attributie" = toekenning).

De patroontabel voor sprites bevat 8 bytes voor elke kleine en 32 bytes voor elke grote sprite. In de sprite-attributietabel kunnen ten hoogste 32 posities op het scherm worden gedefinieerd. Elke gedefinieerde sprite in de attributietabel bestaat uit 4 bytes:

	7 6 5 4 3 2 1 0 Bit
Byte 0	Y-positie
1	X-positie
2	Sprite naam
3	(bits 7, 6–4: leeg) Kleur

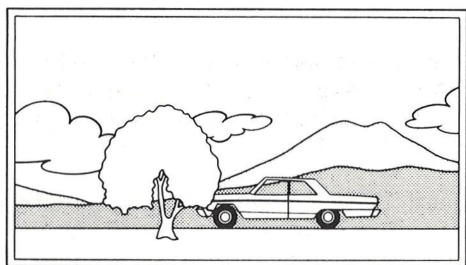
In de eerste byte wordt de Y-positie gezet. Deze waarde, die kan liggen tussen 0 en 255, bepaalt de regel op het scherm. Daarbij hebben de waarden de volgende betekenis:

0...191	geeft het regelnummer op het scherm aan;
208	de sprite is niet zichtbaar op het scherm;
224...255	komt overeen met de schermregels -31...-1; deze waarden maken het mogelijk een sprite boven de bovenste regel op het scherm te plaatsen.

Ook aan de tweede byte, die de X-positie op de regel aangeeft, kunnen waarden van 0 tot 255 worden toegekend. Als bit 7 van de kleurbyte gelijk is aan 1, dan komt de positie van de sprite op de regel overeen met de X-positie minus 32. Is X = 0 en is bit 7 van de kleurbyte gelijk aan 1, dan wordt de sprite in kolom -32 geplaatst. Dit maakt het mogelijk een sprite vóór de eerste kolom te plaatsen.

De Y- en X-positie, aangegeven in de eerste twee bytes, heeft betrekking op het punt geheel linksboven van de sprite.
De sprite die het eerst in de attibutietabel is geplaatst heeft het nummer 0 en daarmee de hoogste prioriteit.

In de volgende afbeelding ziet u een auto die door een landschap rijdt. De boom bestaat uit de sprites 0 en 1, de auto uit de sprites 2...5 en de wolken uit de sprites 6, 7 en 8. Omdat de sprites 0 en 1 voorrang hebben boven de sprites 2...5, zal de boom vóór de auto komen te staan.
De derde byte in de attributietabel is de "naam" van de sprite. Dit is een nummer dat aangeeft waar het patroon van de sprite in de patroontabel kan worden gevonden.
Het eerste adres van de sprite-attributietabel in het videogeheugen is bij tekststand 2 te vinden in de variabele "BASE(8)", bij grafische stand 1 in BASE(13) en bij grafische stand 2 in BASE(18).



ACHTERVLAK

PATROON-
VLAK

SPRITE 31

SPRITE 8

SPRITE 7

SPRITE 6

SPRITE 5

SPRITE 4

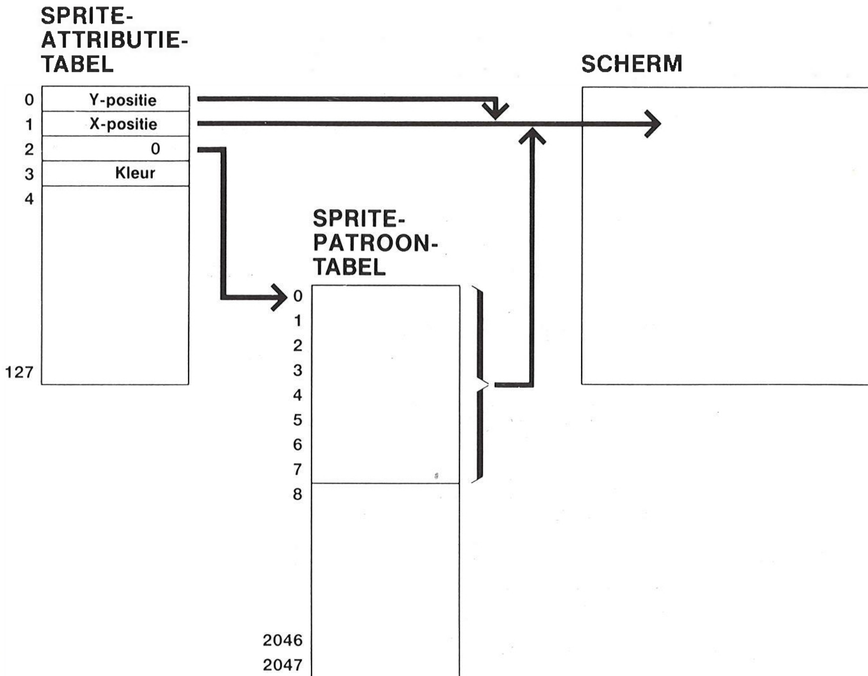
SPRITE 3

SPRITE 2

SPRITE 1

SPRITE 0

Het eerste adres van de sprite-patroontabel in het videogeheugen is bij tekststand 2 te vinden in de variabele BASE(9), bij grafische stand 1 in BASE(14) en bij grafische stand 2 in BASE(19).



Wis het geheugen met het commando NEW en toets het volgende programma in.

```
20 DATA 16,56,68,254,16,16,16,16
30 FOR I=0 to 7
40 READ A:VPOKE(BASE(9)+I),A
50 NEXT I
60 PRINT:PRINT:PRINT
70 PRINT"Een sprite in tekststand
1"
80 VPOKE (BASE(8)+0),40
90 VPOKE (BASE(8)+2),0
100 VPOKE (BASE(8)+3),1
```

```
110 FOR I=0 to 255
120 VPOKE (BASE(8)+1),I
130 NEXT I
140 GOTO 110
RUN
```

Dit voorbeeld laat zien dat het wel degelijk mogelijk is in tekststand 2 sprites te gebruiken door rechtstreeks met het videogeheugen van de VDP te communiceren. Onderbreek nu het programma en toets in: SCREEN 0:WIDTH 40.

6 Registers

De VDP heeft 9 registers tot zijn beschikking. De inhoud van de registers ligt bij MSX-BASIC vast in de variabele VDP(n), waarin n een getal is tussen 0 en 8. Met uitzondering van het laatste register, nummer 8, mag de inhoud van de registers worden veranderd. De registers hebben de volgende betekenis:

Bit	0	1	2	3	4	5	6	7
Register 0	0	0	0	0	0	0	A	D
Register 1	1	R	E	B	C	0	S	M
Register 2	0	0	0	0	**Naamtabel**			
Register 3	**Kleurtabel**							
Register 4	0	0	0	0	0	**Patroontabel**		
Register 5	0	**Sprite-attributietabel**						

Register 6	**0**	**0**	**0**	**0**	**0**	**Sprite-patroon-tabel**

Register 7	**Tekst kleur 1**	**Tekst kleur 2**

Register 8	**T**	**N**	**V**	**5de Sprite**

ABC	000	Tekststand 2
	100	Grafische stand 1
	001	Grafische stand 2
	010	Tekststand 1
D	0	Geen externe invoer naar VDP
	1	Wel externe invoer naar VDP
E	0	Geen interruptie van VDP
	1	Wel interruptie van VDP
R	0	Geen presentatie op het scherm (het scherm toont tekst kleur 2)
	1	Wel presentatie op het scherm.
S	0	Sprites zijn 8 x 8 beeldelementen groot
	1	Sprites zijn 16 x 16 beeldelementen groot
M	0	Sprites zijn niet vergroot
	1	Sprites zijn wel vergroot
T		De T-statusvlag in register 8 wordt aan het einde van elk beeldraster op 1 gezet (dit wil zeggen aan het einde van de laatste beeldlijn in het actieve venster op het scherm). De vlag wordt teruggesteld op 0 nadat het status-

	register is uitgelezen of de VDP extern wordt teruggesteld ("reset").
N	De N-statusvlag in register 8 wordt op 1 gezet als twee of meer sprites geheel of gedeeltelijk samenvallen; dat wil zeggen dat ze ten minste één beeld-element gemeenschappelijk hebben. De N-vlag wordt weer op 0 gezet als register 8 is uitgelezen of als de VDP extern wordt teruggesteld ("reset"). Register 8 moet bij het inschakelen van de computer onmiddellijk worden uitgelezen om er zeker van te zijn dat de N-vlag op nul is gezet. De VDP controleert bij het genereren van een beeldelement alle beeld-elementen op het samenvallen van sprites, onverschillig waar zij zich op het scherm bevinden. Bij de TMS 9929A gebeurt dat elke 20 ms. Als een sprite gedurende zo'n interval over een afstand van meer dan 1 beeldelement wordt verplaatst, is het mogelijk dat meer dan één beeldelement van de sprites samenvalt. Ze kunnen elkaar zelfs helemaal gepasseerd zijn op het moment dat de VDP kijkt of er sprite-elementen samenvallen.
V en 5de sprite	De V-statusvlag in register 8 wordt op 1 gezet als er vijf of meer sprites op een regel (0...192) voorkomen. Hij wordt teruggesteld op 0 als register 8 is uitgelezen en als de VDP extern wordt teruggesteld. Het nummer van de vijfde sprite wordt in de laatste 5 bits van register 8 geplaatst als de V-vlag op 1 wordt gezet. Dit nummer is alleen geldig zolang de V-vlag 1 is.
Tekstkleur 1	De voorgrondkleur in tekststand 1
Tekstkleur 2	De achtergrondkleur in tekststand 1 of

de kleur van de rand in tekststand 2 en in grafische stand 1 en 2.

De registers "Naamtabel", "Kleurtabel" enz. bevatten het eerste adres van de betrokken tabellen.

De tabellen in het videogeheugen mogen door de gebruiker op een andere plaats worden gezet dan door MSX-BASIC is aangegeven, mits de volgende regels in acht worden genomen:

1 De naamtabel moet beginnen op adres 0 of op een adres dat een veelvoud is van 1024.
2 De kleurtabel moet beginnen op adres 0 of op een adres dat een veelvoud is van 64.
3 De patroontabel moet beginnen op adres 0 of op een adres dat een veelvoud is van 2048.
4 De sprite-attributietabel moet beginnen op adres 0 of op een adres dat een veelvoud is van 128.
5 De sprite-patroontabel moet beginnen op adres 0 of op een adres dat een veelvoud is van 2048.

Appendix J De programmeerbare geluidsgenerator

Uw MSX-computer is uitgerust met een programmeerbare geluidsgenerator (Programmable Sound Generator, PSG) met zestien registers. Instructies kunnen met behulp van de instructie SOUND in de vorm van een waarde rechtstreeks in deze registers worden geplaatst. In tegenstelling met de instructie PLAY, blijft bij gebruik van SOUND het geluid voortduren totdat de inhoud van het betrokken register nul wordt gemaakt met een volgende SOUND-instructie.

Bit	7	6	5	4	3	2	1	0
Register 0	Toonnummer kanaal A							
Register 1	0	0	0	0	Toon A grof			
Register 2	Toonnummer kanaal B							
Register 3	0	0	0	0	Toon B grof			
Register 4	Toonnummer kanaal C							
Register 5	0	0	0	0	Toon C grof			
Register 6	0	0	0	Ruistoon				
Register 7	0	0	RC	RB	RA	TC	TB	TA
Register 8	0	0	0	M	Volume A			
Register 9	0	0	0	M	Volume B			
Register 10	0	0	0	M	Volume C			
Register 11	Vormlengte toon							
Register 12	Vormlengte grove toon							
Register 13	0	0	0	0	Vormnummer			

De toonhoogte voor de kanalen A, B en C wordt in de registers 0 tot en met 5 geplaatst. Register 7 bepaalt voor welk kanaal de aangegeven toon of ruis bestemd is. Als de binaire inhoud van register 7 00111110 bedraagt, betekent dit dat de toon alleen via kanaal A wordt weergegeven.
De geluidssterkte voor alle kanalen is vastgelegd in de registers 8, 9 en 10. Als bit M 1 is, is het geluidsvolume het grootst.
De registers 11 en 12 bepalen de lengte van gekozen geluidsvorm die in register 13 staat. Het vormnummer in register 13 heeft dezelfde betekenis als de subcommando'n in de instructie PLAY.
Registers 14, 15 en 16 (niet getekend in de afbeelding) doen dienst als in- en uitgangspoorten voor de kanalen A, B en C.

1 Toonnummer

De toon voor elk kanaal wordt als volgt bepaald:
TP = CF/(16 x TF) CT + T/256 = TP/256

Hierin is:

TP	=	periodetijd van de toon ("Tone Period")
CF	=	klokfrequentie ("Clock Frequency" = 1,99675 MHz)
TF	=	toonfrequentie
T	=	toon (register 0, 2 of 4)
CT	=	grove toon ("Coarse Tone") (register 1, 3 of 5)

Voorbeeld:
Een toon van 440 Hz voor kanaal A wordt als volgt bepaald:

TP = 1.99675 x 10 6/(16 x 440) = 283

CT + T/256 = 283/256

CT = 256/256 = 1

T = 27

Dit betekent: register 0 = 27 en register 1 = 1.

2 Ruistonen

Ruis wordt als volgt bepaald:
NP = CF/(16 x NF) NT = NP

Hierin is:
NP = periodeduur ruis ("Noise Period")
CF = klokfrequentie (= 1.99675 MHz)
NF = ruisfrequentie ("Noise Frequency")
NT = ruistoon ("Noise Tone") (register 6)

Voorbeeld:
Een ruistoon van 30 kHz wordt als volgt bepaald:

NP = (1.99675 x 10 6)/(16 x (30 x 10 3)) = 4

NT = 4

Dit betekent: register 5 = 4

3 Lengte van de vorm

De vormlengte wordt als volgt bepaald:
SP = CF/(256 x SF) CT + T/256 = SP/256

Hierin is:
SP = periodetijd vorm ("Shape Period")
CF = klokfrequentie (= 1.99675 MHz)
SF = vormfrequentie ("Shape Frequency")
T = vormlengte van de toon (register 11)
CT = vormlengte van de grove toon (register 12)

Voorbeeld:
Een vormfrequentie van 0,5 Hz wordt als volgt bepaald:

SP = (1.99675 x 10 6)/(256 x 0.5) = 156000

CT + T/256 = 15600/256

CT = 15360/256 = 60

T = 240

Dit betekent: register 11 = 240 en register 12 = 60

4 Programmavoorbeeld

Wis het geheugen met het commando NEW en toets het volgende programma in:

```
10 FOR I=0 TO 13
20 SOUND I,0
30 NEXT I
40 SOUND 7,&H3E
50 SOUND 8,&H0F
60 FOR I= &H10 TO &HF0
70 SOUND 0,I
80 NEXT I
90 SOUND 6,&H0F
100 SOUND 7,&H07
110 SOUND 8,&H10
120 SOUND 9,&H10
130 SOUND 10,&H10
140 SOUND 12,&H40
150 SOUND 13,&H00
RUN
```

Met behulp van de FOR....NEXT-lus in de regels 10 tot en met 30 worden alle registers op 0 gezet. In regel 40 wordt alleen via kanaal A een toon toegelaten en in regel 50 wordt de geluidssterkte van kanaal A bepaald.
In de FOR...NEXT-lus van regel 60 tot en met 80 worden achtereenvolgens de tonen voor kanaal A geplaatst.
In regel 90 wordt de ruistoon bepaald.
In regel 100 is aangegeven dat de ruis moet worden weergegeven via de kanalen A, B en C.
In regel 110 tot en met 130 is de maximum-geluidssterkte gegeven voor alle drie de kanalen.
In regel 140 is de vormlengte bepaald.
In regel 150 wordt tenslotte de vorm van de ruis bepaald.

5 De zee

Het volgende programma bootst het geluid van de zee na:

```
10 FOR I=0 TO 13
20 SOUND I,0
```

```
30 NEXT I
40 SOUND 6,&H10
50 SOUND 7,&H07
60 SOUND 8,&H10
70 SOUND 9,&H10
80 SOUND 10,&H10
90 SOUND 12,&H40
100 SOUND 13,&H0E
RUN
```

Alle geluiden die u hoort worden geproduceerd door de geluidsgenerator. Nadat het programma is uitgevoerd, keert MSX-BASIC terug naar de directe stand, maar het geluid gaat door. Dit stopt pas als de registers 6 en 13 nul worden gemaakt of als u tegelijk de toetsen CTRL en STOP indrukt.



PHILIPS